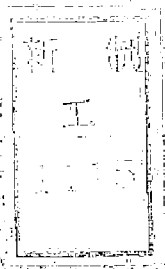
Studies on Robust Language and Dialogue Processing for Spoken Dialogue Systems(Dissertation_全文)

AUTHOR(S):

Araki, Masahiro



Studies on Robust Language and Dialogue Processing for Spoken Dialogue Systems

Masahiro ARAKI

December 1997

Studies on Robust Language and Dialogue Processing for Spoken Dialogue Systems

Masahiro ARAKI

December 1997

Abstract

In spoken dialogue systems, robust language processing for spontaneous speech understanding and robust dialogue processing for achieving user goal are inevitable. Previously, research of speech recognition and research of natural language understanding were done independently. At first glance, it seems to be no problem to combine these two technologies, because the purpose of speech recognition is to identify spoken object (word or sentence) from the input speech, and one of the purposes of the natural language understanding is to make some semantic representation from the sentence. However, the demonstration dialogue system which simply concatenate these two technologies did not work well if the input is not correct utterance, or the recognition error occurs.

In this thesis, we describe robust language processing and robust dialogue processing for spoken dialogue systems. First, we explain the robust language processing method, that is, keyword-driven parser, which uses path analysis of the semantic network. The keyword-driven parser can generate partial semantic representation toward the noisy input. Next, we propose a cognitive process model of spoken dialogue, which specifies the cognitive process of whole dialogue understanding process, the interaction between understanding process and dialogue management process, and the recovering method from input errors into this cognitive process. Such robustness should be evaluated in pseudo real environment for spoken dialogue systems, that is, interactive environment under communication errors. We implemented system-to-system dialogue evaluation environment with linguistic noise, and showed the effectiveness of proposed dialogue model.

In chapter 1, we clarify our position with describing general framework of spoken dialogue systems and describing the previous approach to robust language / dialogue processing.

In chapter 2, we present a keyword-driven speech parser as a robust language processing. Generally speaking, previous robust parsing methods are divided into two groups: grammar-based approach (it is a kind of *theory-based technique*) that generates all possible

hypotheses corresponding to deletion, insertion and substitution of words, and pragmatics-based approach (it is a kind of *task-oriented technique*) that uses sentence templates fixing the roles of content words. In our approach, the seeds of the utterance analysis are words in the same way as pragmatics-based approach. In combining these seeds words into partial semantic representation, we use the path description of semantic network and partial grammar which is a set of rules for Japanese phrase *'bunsetsu'*. We call this method as semantic-based approach. In addition, dialogue level predictions can be used in our method by pruning the search space in activated subnetwork. By this method, we realize a semantic analyzer that achieve 69.3% in semantic understanding rate, and 87.5% dialogue continuation rate (less than one error contained in keyword, except for verb).

In chapter 3, we propose a cognitive process model of spoken dialogue. In order to make an interactive dialogue system, we need two management processes: one is understanding process which manages the subprocess of utterance understanding through response generation; the other is dialogue management process which aggregates the utterances to the discourse segment, manages focus and intentions of dialogue. Furthermore, in applying the model to spoken dialogue systems, we have to deal with input errors caused by speech recognition errors. Our model specifies the cognitive process of whole dialogue understanding process and stipulates the interaction between understanding process and dialogue management process. We also specify the recovering method from input errors into this cognitive process. Therefore, our model is suitable for implementing cooperative spoken dialogue systems.

In chapter 4, we propose an evaluation environment for robust language / dialogue processing under interactive situation. We use this environment for evaluating proposed robust processing method. In robust language processing, the parameter can be varied to make precision higher, that means restraining only plausible output, or to make recall higher, that means generating the output anyway. On the other hand, in robust dialogue processing, the dialogue strategy which manages the communication error affects the task achievement rate or redundancy of dialogue. In order to determine such parameters, we need interactive dialogue situation. The recorded data cannot be used anymore for this purpose. In evaluating our system's robustness to recognition errors or ill-formed sentences in spoken dialogue systems, we designed linguistic noisy channel in system-to-system automatic dialogue and establish evaluation methodology such interactive systems. In this environment, we examined the effectiveness of our robust processing methods.

In chapter 5, we describe conclusions of this thesis and future works.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Towards robust spoken dialogue systems

Speech is in daily use as a natural medium of communication. In using speech as input device of computer, it provides friendly, hands-free and location-independent input medium. Recent advances in speech recognition, natural language processing, and dialogue understanding have made it possible to build spoken dialogue systems for a wide variety of applications. There are several demonstrated spoken dialogue systems such as air travel information service, sight seeing guide, database interface, telephone directory etc. At first glance, these systems work well to user's request. But actually, many of these systems cannot continue dialogue when user deviate their dialogue pattern. In addition, some of these systems are fragile to speech recognition errors.

In order to implement more useful and friendly spoken dialogue systems, it is necessary to develop robust language processing for spontaneous speech understanding and a robust dialogue processing for achieving user goal under communication errors.

Previously, research of speech recognition and research of natural language understanding were done independently. At first glance, it seems to be no problem to combine these two technologies, because the purpose of speech recognition is to identify spoken object (word or sentence) from the input speech, and one of the purposes of the natural language understanding is to make some semantic representation from the sentence. The simple integration of these method presupposes that there is no error in interface. That is to say, speech recognition system and natural language understanding system are assumed to share the same node in language level of Figure 1.1.

Some demonstrated spoken dialogue systems were implemented which appear to integrate speech recognition and natural language understanding using such simple concate-

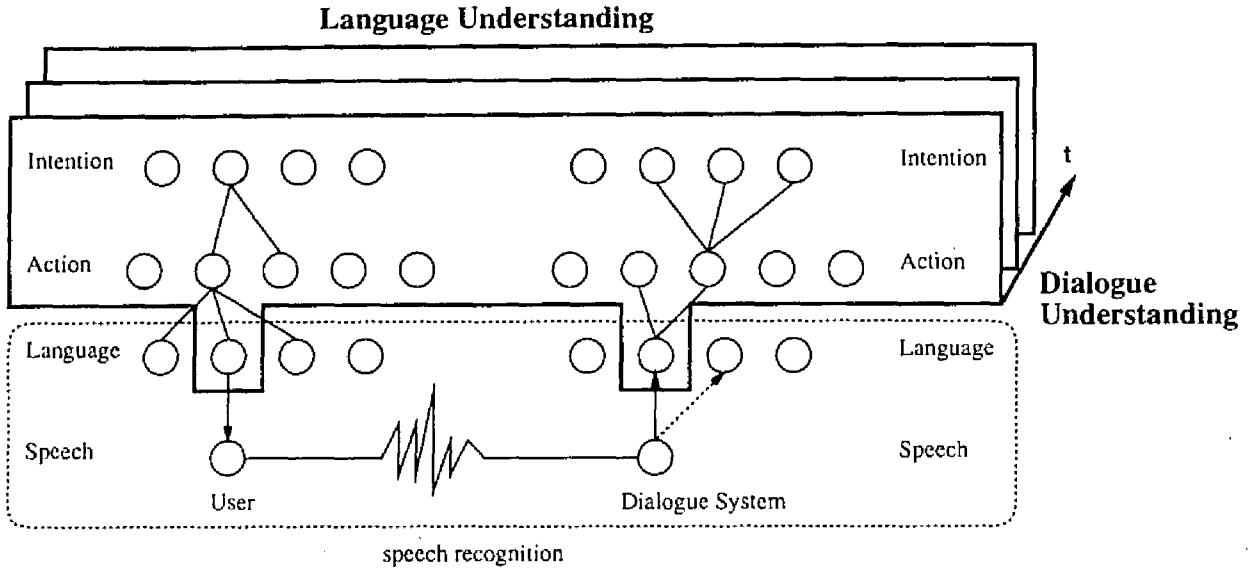


Figure 1.1: Relation between speech recognition and natural language understanding

nation. Such demonstration system did not work well if the input is not correct utterance or if recognition error occurs. In order to avoid breakdown of conversation, the task of spoken dialogue system was selected as the number of input utterance type is small and/or interaction completes only one or a few turns.

There were little work concerning how to manage dialogue and how to recover misrecognition in case there exist errors in spoken dialogue systems, that is to say, how to resolve the problem in connection with misrecognition of speech recognition. The problem can be illustrated as user's utterance is mapped wrong node as system's recognized result in language level of Figure 1.1.

Therefore, the recovering problem can be divided into two levels:

1. how to recover errors using semantic/syntactic knowledge on the process of mapping language level to action level (see Figure 1.2),

2. how to recover errors using dialogue/task knowledge on the process of mapping action level to intention level (see Figure 1.3).

The method for resolving the first problem can be evaluated using recoded data by its understanding precision. On the contrary, the method for resolving the second problem can not be evaluated easily because it includes interactive aspect. So, we have to develop the evaluation environment for the robustness of interactive systems.

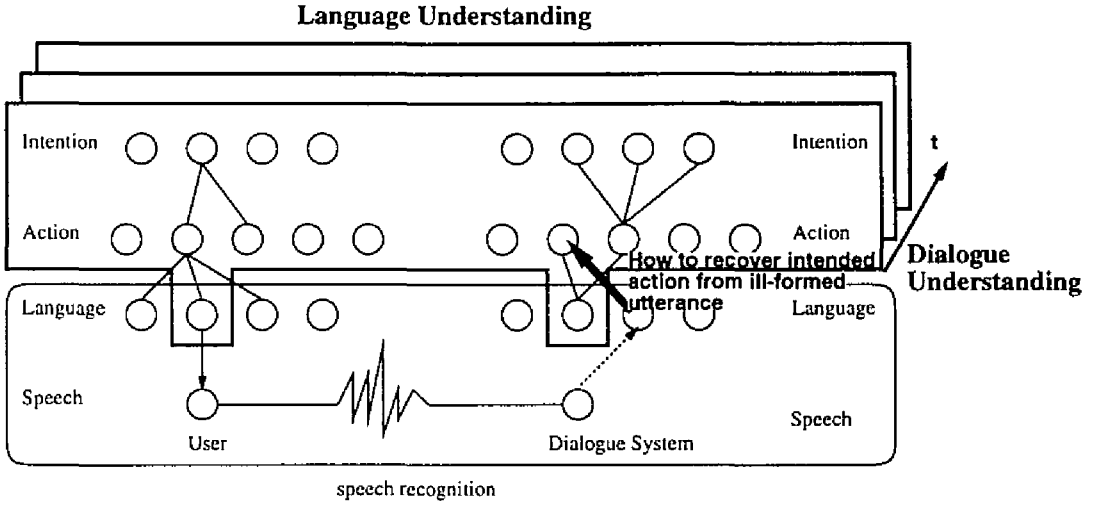


Figure 1.2: Recovering from errors on the process of mapping language level to action level

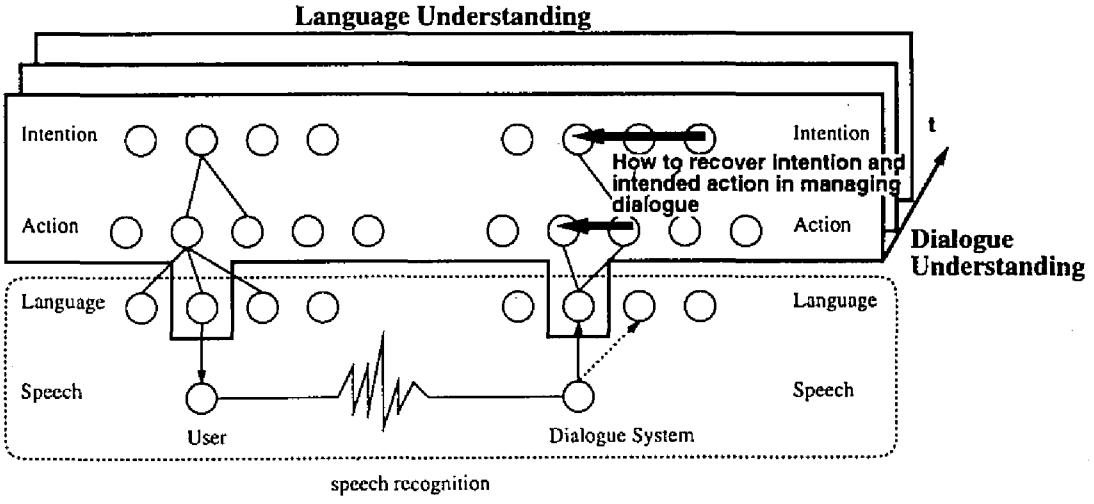


Figure 1.3: Recovering from errors on the process of mapping action level to intention level

In this thesis, we describe our approaches of robust language processing and robust dialogue processing for spoken dialogue systems. Also, the generality and the effectiveness are examined under the interactive evaluation environment.

Toward the first problem, we developed the robust language processing method using path analysis of the semantic network, that can generate partial semantic representation toward the noisy input.

Toward the second problem, we propose a cognitive process model of spoken dialogue, which specifies the cognitive process of whole dialogue understanding process, the interaction between understanding process and dialogue management process, and the recovering method from input errors into this cognitive process.

Toward the third problem, we designed an interactive environment under communication errors, that is, system-to-system dialogue evaluation environment with linguistic noise that can show the effectiveness of proposed robust language processing and dialogue model.

1.2 Basic architecture of spoken dialogue system

As a preliminary of following discussion, we overview a basic architecture of spoken dialogue systems based on our personal scheduling system, in which the robust processing systems presented in this thesis are to be integrated.

In our personal scheduling system, we use *xcalendar*, which is an application program on X-Window, as a schedule database and a graphical user interface. Sample sentences in this task are concerning database access (assert, query, modify) and system control (reply to system statement, closing session, etc.). The screen image of the system is shown in Figure 1.4.

Our prototype system is being build with some subsystem — acoustic component, lattice generator, A* parser, language processor, dialogue manager, database manager, and speech synthesizer —. The overall structure of spoken dialogue system is shown in Figure 1.5. The front-end level of the system, i. e. acoustic component, lattice generator, A* parser are not the theme of this thesis. We use commercial speech synthesizer AI-talk2 by SANYO.

Figure 1.4: Screen image of spoken dialogue system

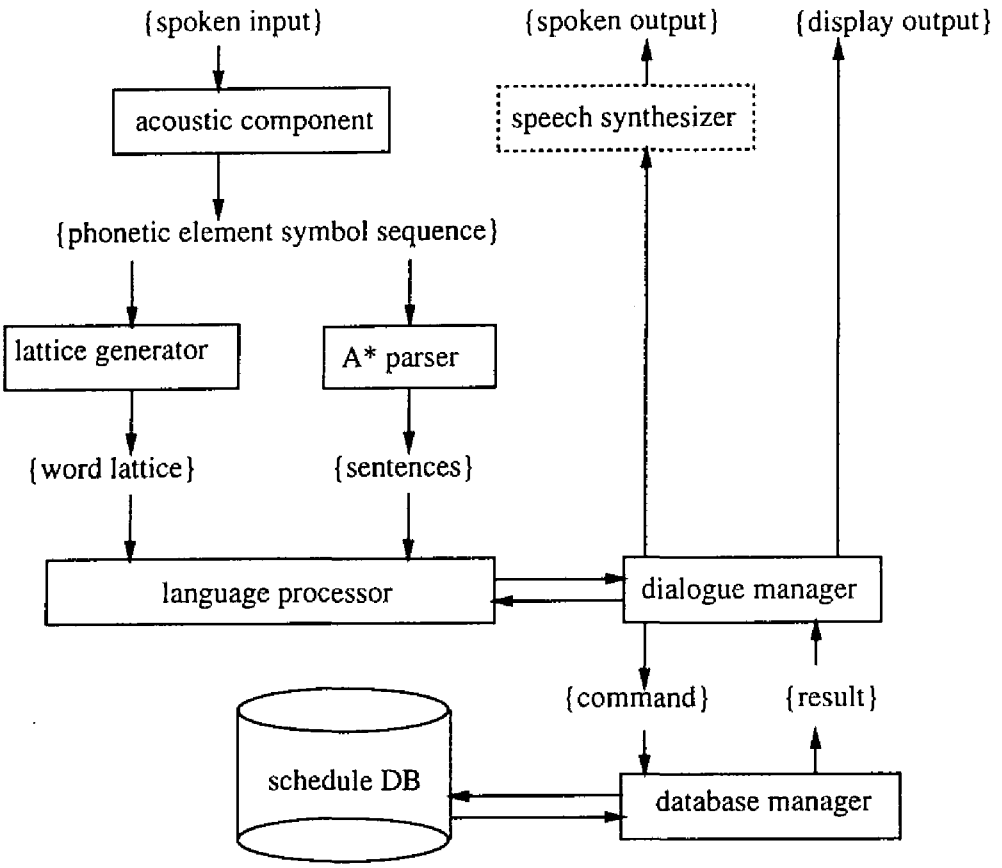


Figure 1.5: Overall structure of spoken dialogue system

1.2.1 Acoustic component

The input of acoustic component is speaker-independent continuous speech. The basic algorithm of this component is following:

1. Every frame of input speech is analyzed by 26-th order LPC analysis to make pattern vector.

2. Bayes classifier is applied to get a phonetic element symbol sequence.

The output of this module is a phonetic element symbol sequence.

1.2.2 A* parser

In parsing approach, we use word-pair constraint as heuristics at first pass, and A* admissible right-to-left search at second path [1]. Heuristics ($\hat{h}(n)$) estimates the score of

the remaining unsearched part. On the other hand, right-to-left search calculates the score of the searched part ($g(n)$). Therefore, the evaluation function for a hypothesis n is formulated as follows.

$$\hat{f}(n) = g(n) + \hat{h}(n)$$

The A* algorithm develops the search proceeding the high score hypotheses. The output of the A* parser is N-best sentence hypotheses which is passed to the language processor.

1.2.3 Lattice generator

The lattice generator uses spotting model with heuristic language model [2]. The heuristic spotting algorithm consists of following two phases:

1. the heuristic model is applied to the whole part of an input speech,

2. the evaluation function of the spotted word is obtained by the sum of left context heuristic score, right context heuristic score and the score of word model.

The output of the lattice generator is word lattice which is also passed to the language processor.

1.2.4 Language processor

The input of the language processor is word lattice (the output of the semantic first approach that is based on word spotting) or N-best sentence hypotheses (the output of the syntactic approach implemented by A* search of trellis space).

The requirement of this module is to decide the grammatical structure of the input utterance and extract semantic representation. The baseline technique of this language processor is parsing by semantic grammar, in which non-terminal symbol of the grammar is semantic category.

The output of this language processor is semantic expression which is independent to dialogue context.

1.2.5 Dialogue manager

The input of the dialogue manager is context independent semantic expression of the input utterance.

The dialogue manager plays following three roles in this setting.

1. Translation from semantic representation of the user utterance to a database access command

2. Producing answer expression to the user utterance referring schedule database

3. Prediction of the next user utterance by marking and partitioning the network

In order to measure the effect of information integration in spoken language processing, e. g. syntax, semantics and dialogue structure, we modeled dialogue by automaton and examine the understanding rate under the situation of each state.

The role of Dialogue management subsystem is to answer to user utterance referring schedule database and to make the prediction by operation on network. Some automata are prepared according to the user goal of Dialogue. Only one goal in a Dialogue is assumed. The automaton dialogue model is shown in Figure 1.6.

The predictions of the next utterance of the user are made by the knowledge of user goal, Dialogue structure and topics. We use these knowledge sources to mark nodes that will be preferred in path analysis and to partition the network to limit connectable word group.

1.3 Outline of the thesis

In chapter 1, we clarify our position with describing general framework of spoken dialogue systems and describing the previous approach to robust language / dialogue processing.

In chapter 2, we present a keyword-driven speech parser as a robust language processing. Generally speaking, previous robust parsing methods are divided into two groups: grammar-based approach (it is a kind of *theory-based technique*) that generates all possible hypotheses corresponding to deletion, insertion and substitution of words, and pragmatics-based approach (it is a kind of *task-oriented technique*) that uses sentence templates fixing the roles of content words. In our approach, the seeds of the utterance analysis are words

in the same way as pragmatics-based approach. In combining these seeds words into partial semantic representation, we use the path description of semantic network and partial grammar which is a set of rules for Japanese phrase *'bunsetsu'*. We call this method as semantic-based approach. In addition, dialogue level predictions can be used in our method by pruning the search space in activated subnetwork. By this method, we realize a semantic analyzer that achieve 69.3% in semantic understanding rate, and 87.5% dialogue continuation rate (less than one error contained in keyword, except for verb).

In chapter 3, we propose a cognitive process model of spoken dialogue. In order to make an interactive dialogue system, we need two management processes: one is understanding process which manages the subprocess of utterance understanding through response generation; the other is dialogue management process which aggregates the utterances to the discourse segment, manages focus and intentions of dialogue. Furthermore, in applying the model to spoken dialogue systems, we have to deal with input errors caused by speech recognition errors. Our model specifies the cognitive process of whole dialogue understanding process and stipulates the interaction between understanding process and dialogue management process. We also specify the recovering method from input errors into this cognitive process. Therefore, our model is suitable for implementing cooperative spoken dialogue systems.

In chapter 4, we propose an evaluation environment for robust language / dialogue processing under interactive situation. We use this environment for evaluating proposed robust processing method. In robust language processing, the parameter can be varied to make precision higher, that means restraining only plausible output, or to make recall higher, that means generating the output anyway. On the other hand, in robust dialogue processing, the dialogue strategy which manages the communication error affects the task achievement rate or redundancy of dialogue. In order to determine such parameters, we need interactive dialogue situation. The recorded data cannot be used anymore for this purpose. In evaluating our system's robustness to recognition errors or ill-formed sentences in spoken dialogue systems, we designed linguistic noisy channel in system-to-system automatic dialogue and establish evaluation methodology such interactive systems. In this environment, we examined the effectiveness of our robust processing methods.

In chapter 5, we describe conclusions of this thesis and future works.

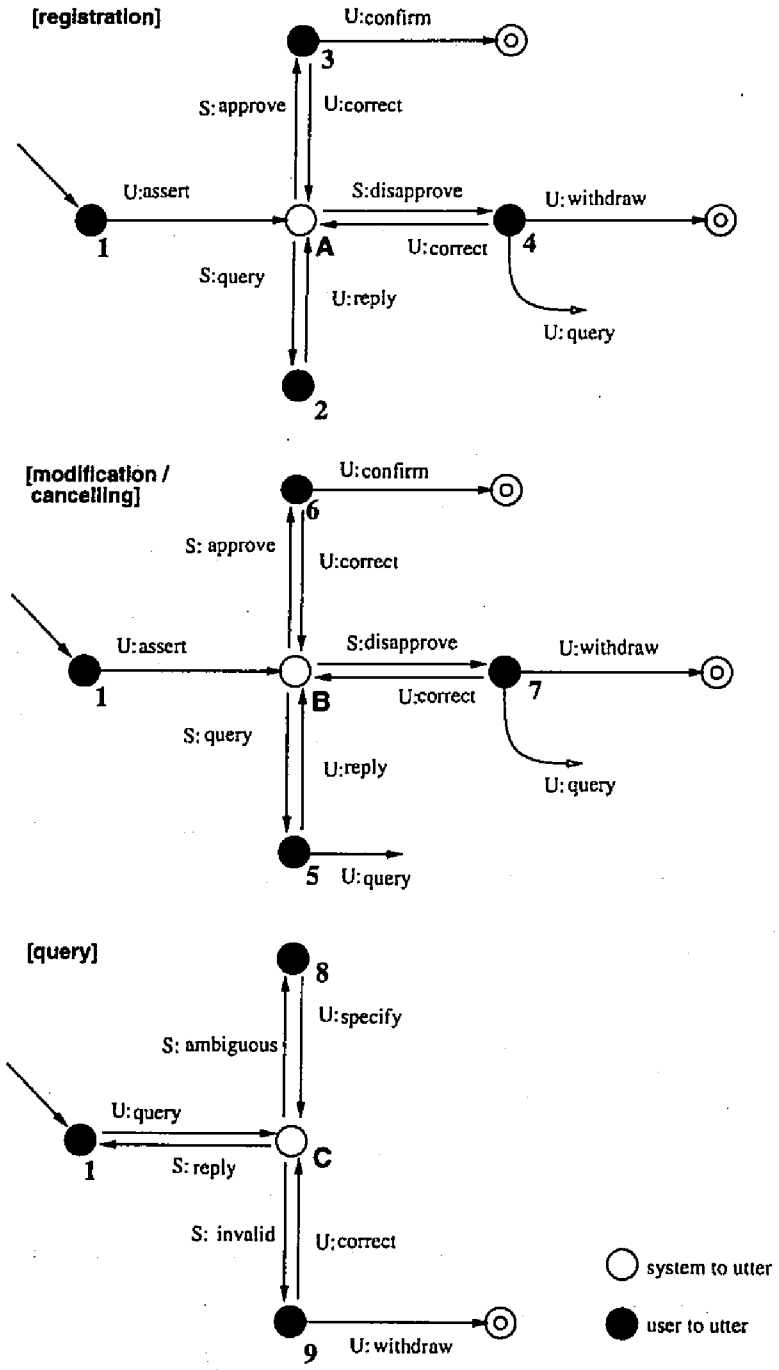


Figure 1.6: Automaton dialogue model

Chapter 2

Keyword-driven Parser for Robust Language Processing

2.1 Introduction

The input of spoken dialogue systems include various ambiguity and uncertainty of user's input, such as uncertainty of speech recognition results, syntactic and semantic ambiguity, ill-formed utterances and uncertainty of user's intention.

In this chapter, we present a keyword-driven speech parser as a robust language processing. Generally speaking, previous robust parsing methods are divided into two groups: grammar-based approach (it is a kind of *theory-based technique*) that generates all possible hypotheses corresponding to deletion, insertion and substitution of words, and pragmatics-based approach (it is a kind of *task-oriented technique*) that uses sentence templates fixing the roles of content words. In our approach, the seeds of the utterance analysis are words in the same way as pragmatics-based approach. In combining these seeds words into partial semantic representation, we use the path description of semantic network and partial grammar which is a set of rules for Japanese phrase *'bunsetsu'*. We call this method as 'semantic-based approach.' In addition, dialogue level predictions can be used in our method by limiting the search space in activated subnetwork.

2.2 Survey of robust language processing

Previous researches concerning robust language processing are divided in two ways: one is a theory-based method, the other is task-oriented method. Theory-based method generally deals with a slight derivation from grammar [3], [4], [5]. On the other hand, task-oriented method typically uses pragmatic template for language processing [6], [7].

Recently, as corpus based language processing became popular, another approach for robust language processing are proposed, that is, probabilistic method [8], [9], [6], [10].

In this section, we overview these three approaches for robust parsing in order to locate our keyword-driven approach as a current solution.

2.2.1 Theory-based method

Theory-based technique generally deals with a slight derivation from grammar[3], [4], [5]. Each derivation pattern and its recovering method are written also in rule.

Defect of theory-based method

Most of such methods were annoyed by the large amount of computational cost because each processing slice has its hypotheses and these hypotheses are multiplied with next hypotheses. As a processing continues, the amount of computational cost (or number of hypotheses) increase exponentially. In order to avoid this problem, many of theory-based technique assumed the number of errors in small number (one or a few). It may be used in the processing of OCR recognized sentences. But, it in not practical for spoken dialogue systems.

2.2.2 Task-oriented method

Task-oriented technique typically uses pragmatic template for language processing [6], [7]. These methods are classified into flame-based method and knowledge-integration method.

Flame-based technique

In studies on spontaneous speech understanding, the major approach is frame-based method. Ward developed Phoenix system [11] with flexible parsing that combines frame-based semantics with a semantic phrase grammar. In frame-based method, semantic representation is limited by prepared frames. These are task dependent.

Jackson et al. proposed the combination method [7]. The template matcher works after when analytical parser cannot produce a complete analysis. But the judgment of the "fail of analysis" is difficult.

Knowledge-integration technique

Speech understanding system requires the use of various knowledge sources to limit the search space. Especially in Dialogue system, these knowledge sources must be used at parsing stage, because they should deal with spontaneous speech. Spontaneous speech often contains filled pauses, restarts, stutters, etc. To extract the meaning from such distorted utterance, the parser needs strong constraints especially based on semantic knowledge and Dialogue-level knowledge.

Recently, there have been several studies concerning the use of Dialogue-level knowledge or spontaneous speech understanding.

MINDS system [12] transforms Dialogue-level predictions into word networks according to Dialogue phase. But the correspondences between concepts and sentence templates are not always apparent. And this system basically based on ATN method, then it will encounter difficulties processing spontaneous speech.

Defect of task-oriented method

Task-oriented method are too pragmatic to apply other task domain. In addition, some of task-oriented method uses heuristic scores in order to determine which template or set of integrated hypothesis is correct. However, such heuristic scores are tuned for its task and are not validated in probabilistic theory.

2.2.3 Probabilistic method

Many probabilistic methods are developed for various spontaneous phenomena, such as probabilistic parsing [8], resolving structural ambiguity [9], semantic processing [6], [10] etc. In order to deal with various ambiguity and uncertainty by integrated manner, we need a framework of probabilistic reasoning. Bayesian network formalism is suitable method to this problem.

Bayesian network approach

Bayesian network is a kind of probabilistic causal network [13]. Each node represents a random variable, that is a value of a proposition. In this chapter, random variable is a binary variable, that is true of false. Each link represents a kind of causal relationship. A certainty measure is assigned to each node that is consistent with the axioms of probability

theory. Its computational cost for updating certainty measure is proportional to the longest path in the network. Because Bayesian network propagates evidential message bidirectional, it can deal with multiple evidence inputs. Then Bayesian network is suitable for treating uncertainty in natural language processing [14].

We regard utterance understanding as dynamic construction of Bayesian network. We call this network Conversational Space (CS). The input of CS is phrase hypothesis that is a result of phrase spotting module. Phrase hypothesis is represented a node with a spotting score as its certainty measure. Some classes of linguistic instances are inferred from this evidence. A proposition that states an existence of instance of conceptual class, utterance type class, action type class is inferred by network expanding procedure.

There are three types of nodes in CS.

- **phrase node** shows the presence of phrase hypothesis. It represents a probabilistic proposition. Its probability is a score of phrase spotting.
- **instance node** is a type of (inst *instance class*) that indicates that an *instance* of a *class* appears in conversation.
- **slot-filling node** is a type of (= (*slot-name instance1*) *instance2*), that indicates *instance2* occupies the *slot-name* slot of *instance1*.

instance node and slot-filling node follow Charniak's definition [15]. Basically, arcs represent a causal relationship. Conditional probability matrix is attached at the each arc. This matrix shows the conditional probability of random variables of two nodes.

There are three types of instance node : concept node, utterance type node and action node. Concept node is expressing a existence of an entity, that is, instance of a concept in conversation. The knowledge source of concept node is a set of pair of phrase template and corresponding concept. Utterance type node is expressing a hypothesis of the type of utterance is made. Table 2.1 [16] shows prepared utterance type.

Each utterance type has slots of concepts and their prior probability. The probability reflects the similar notion of required case and selectional case of case grammar formalism [17]. It is used for conditional probability matrix of the link between utterance type node and concept node. Action node is a top hypothesis node of CS. That corresponds to the leaf node of PSS. It is connected to utterance type node as the same manner as another instance node.

Table 2.1: Utterance type (part)

type	node name	definition
response+	res+	positive response
response-	res-	negative response
acknowledge	ack	acknowledge partner's utterance
confirmation	conf	confirm something
request	req	request something
reject	reject	reject partner's proposal or request
questionref	qref	ask unsettled value to partner
questionif	qif	ask yes or no to partner
inform	inform	give information to partner, or not included above

Each instance node is connected by way of slot-filling node. Only phrase node is dynamically connected to concept node. General idea of CS is shown in Fig. 3.14.

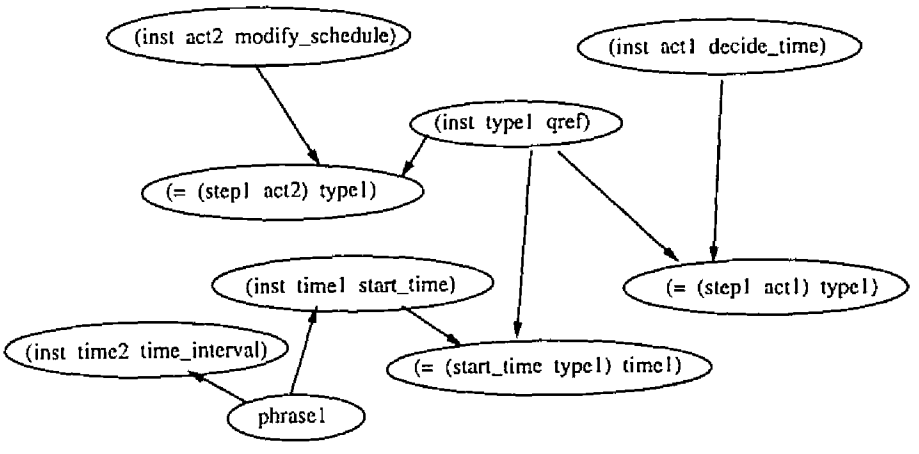


Figure 2.1: Example of probabilistic network

The analysis process in CS is (1) to add nodes and arcs when new instances or relations are brought in, and (2) to decide the most plausible utterance type node at the end of the input. The method of deciding the most plausible utterance type node is based on belief updating method of Bayesian network [13]. The network construction procedure is shown in Fig 2.2.

```
network_construction({p_i})        /* {p_i}: phrase hypotheses */
begin
  for i = 1 to n
    begin
      create_concept_node(p_i, {c_i});
      if {c_i} is_connectable_to {U}    /* {U}: utterance type hypotheses */
        then make_link({c_i}, {U})
        else create_utterance_type_node({c_i}, {u_i}); make_link({c_i}, {u_i})
      endif
    end;
  if |{U}| > 1 then winner({U}, u_win)    else u_win = {U} endif;
  if u_win is_connectable_to {A}    /* {A}: action type hypotheses */
    then make_link(u_win, {A})
    else create_action_type_node(u_win, {a_i}); make_link(u_win, {a_i})
  endif
end
```

Figure 2.2: Network construction procedure

Example of analysis by Baysian network

Here, we explain a processing method in CS and show the example of analysis. We assume sample sentence as "*asu taiwa guruupu no meNbaa no tsugou no ii jikaN ha* " Even though verb phrase is omitted in original Japanese sentence, it may translated into "Tell me an available time of dialogue group members tomorrow." At the same time, though somewhat unnatural. it is taken an inverted sentence in certain situation, as "Dialogue group members are free tomorrow."

We assume the result of phrase spotting is passed to meaning understanding step from first phrase to the last phrase. Each time when meaning understanding step get a phrase hypothesis, corresponding phrase node and concept node, such as (*phrase_id*) and (inst *content-of-phrase concept*), are created. If several concepts can correspond to the content of phrase, instance nodes are created for each concepts. Also if several phrase hypothesis exist at the same segment, each corresponding phrase nodes are generated and assigned normalized score of phrase spotting. For the sake of simplicity, we assume that phrase spotting give the correct result.

If new concept node appears in CS, corresponding utterance type node and slot-filling node are created in CS. New nodes are hypothesis that are based on the evidence of the presence of the concept. Conditional probability matrix is determined by reflecting the

similar notion of required case (with high probability) and selectional case (with rather low probability). The resulting CS of first phrase "*asu* (tomorrow)" is shown in Fig. 2.3.

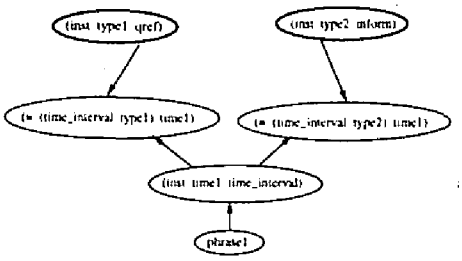


Figure 2.3: Bayesian network parsing (1)

Next phrase "*taiwa guruupu no menbaa no* (of dialogue group members)" makes another concept node. If there exists an empty slot of existing utterance type node, and the new concept is adaptable to this empty slot, then slot-filling node is created and linked to new concept node. If no utterance type node is suitable to the new concept, new utterance type node and slot-filling node are created. It is the same procedure as for the first phrase.

The last phrase "*tsugou no ii jikaN ha* (available time)" also makes the corresponding nodes. A snapshot after processing the last phrase is shown in Fig. 2.4.

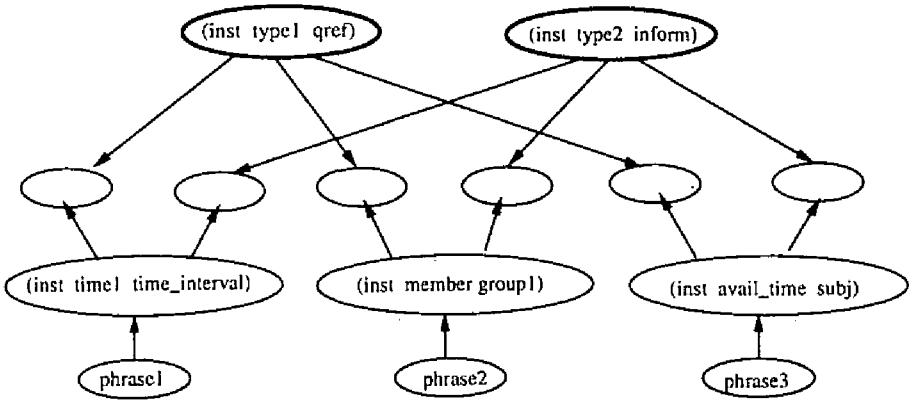


Figure 2.4: Bayesian network parsing (2)

At the end of input, there exist two utterance type nodes in our example. In order to make system's response, the winner of hypothesis of utterance type must be decided. It is done by belief updating method of Bayesian Network [13]. The prior probability

is the appearance probability of each utterance type at the beginning of dialogue. The evidences are the score of phrase spotting.

When most plausible utterance type node is decided, action node, such as (inst *action-id action*), is introduced in CS that have slots of the utterance type. This process roughly means make correspondence to surface utterance type to intended action.

Defect of probabilistic method

Undoubtedly, probabilistic method achieved high performance in syntax level processing, e.g. part of speech tagging in corpus. But, a technique which can deal with semantic information have not established yet. In addition, for such purpose, we need large scale corpus which have already tagged semantic information. Unfortunately, such corpus have not exist yet.

However, as the advantage of probabilistic method is apparent, we should design robust parser as a manner of dealing with probabilistic information.

2.3 Keyword-driven approach

Considering the defects of previous approach, we set our goal as "understanding spontaneous speech using Dialogue-level knowledge source". In this chapter, we describe a method to extract the meaning from a user's utterance by keyword-driven parser that uses network-based integrated knowledge.

The advantage of our keyword-driven method has two aspects. One is a realization of semantic oriented island-driven parser that can parse spontaneous speech. The other aspect is introducing Dialogue-level knowledge without any transformation.

2.3.1 Outline of keyword-driven parser

The structure of keyword-driven parser is shown in Figure 2.5. The input of this parser is a word lattice and the output is a semantic representation of the user utterance closely related to database access command.

2.3.2 Basic algorithm

The parsing algorithm is presented below.

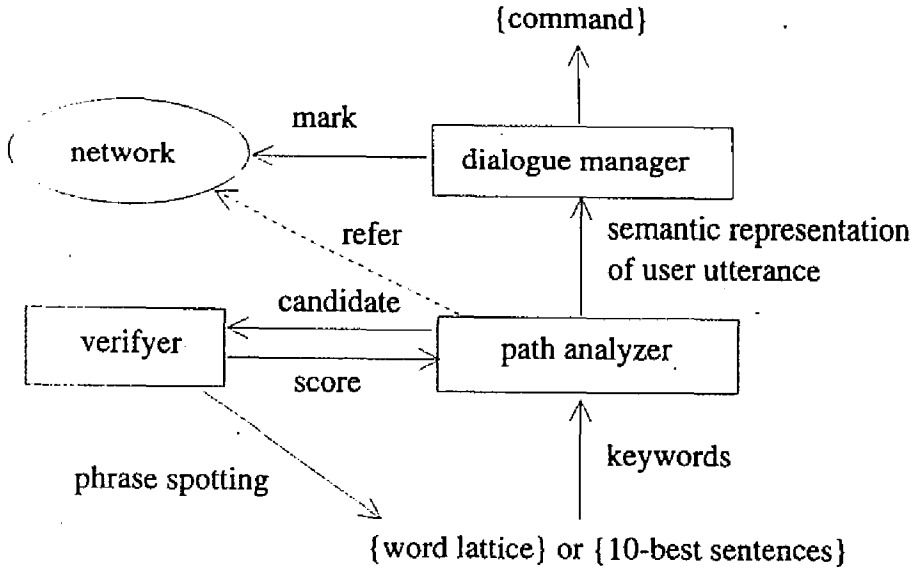


Figure 2.5: Structure of keyword-driven parser

1. Pick up two keywords from word lattice that do not overlap and have the highest score.

2. Make a meaning hypothesis by network path analysis between two keywords.

3. If there are some 'limit' or 'attribute' relation in meaning hypothesis, local syntax check is done.

4. Repeat step 5 to 6 until termination conditions are satisfied.

5. Pick up the next keyword that does not overlap to already picked up words and is not conflict to meaning hypothesis and has high score.

6. Reconstruct the meaning hypothesis by adding the new relation of the picked up word and its neighbor words. If needed, syntax check is done.

2.3.3 Component of keyword-driven parser

In our system, multi-level knowledges (syntax, semantics and pragmatics) are represented in network. The network is constructed by nodes(keyword, word concept, semantic case,

sentence) and arcs (is-a, attribute, limit, element-of). For example, "is-a" relationship between keyword and its class(concept). Figure 2.6 shows a part of the network.

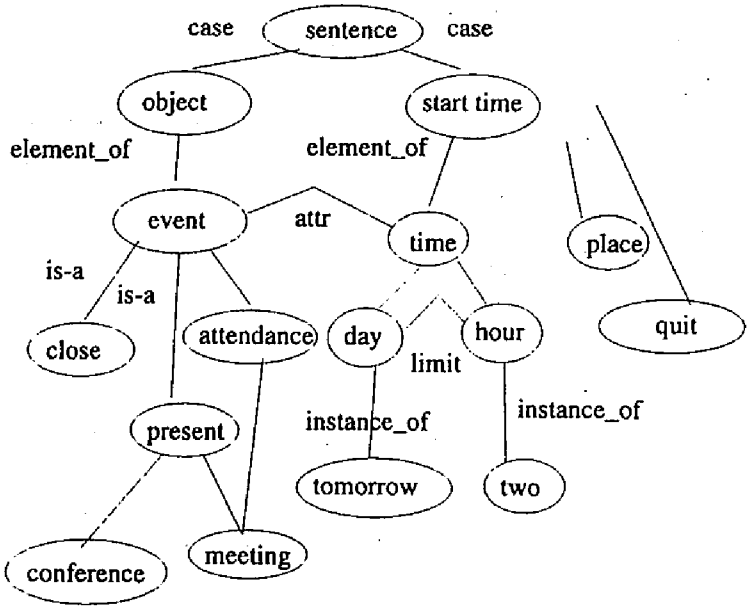


Figure 2.6: Part of the semantic network

The network is constructed by following elements.

Nodes

- keyword
- word concept
- semantic case
- sentence

Arcs

- "is-a" relationship between keyword and its class(concept)
- "attribute" relationship between word concepts
- "limit" relationship between word concepts
- "element-of" relationship between semantic case of sentence and word concept

- "case" relationship between sentence type and semantic case

Spontaneous speech often contains a number of phenomena that strict parser can hardly deal with. In this Keyword-driven parser, filled pauses and restarts cause no problem because the gap of keywords is permitted at some rate. Even if unknown or mispronounced words are included, semantic representation can be constructed only by recognizing words. The lost information should be complemented in successive Dialogue.

2.4 Example of the system behavior

In this section, we describe in more detail our current implementation of the approach outlined above. We assume that a word lattice shown in Figure 2.7 is the input to our parser. The user's utterance is "*Etto..raisyuu no, Etto kayoobi kara, mokuyoobi made, Nagoya ni shuttyou shimasu*" The content of this utterance is "I will make a business trip to Nagoya from Tuesday to Thursday next week." "*Etto*" is one of filled pauses and a comma means short pause.

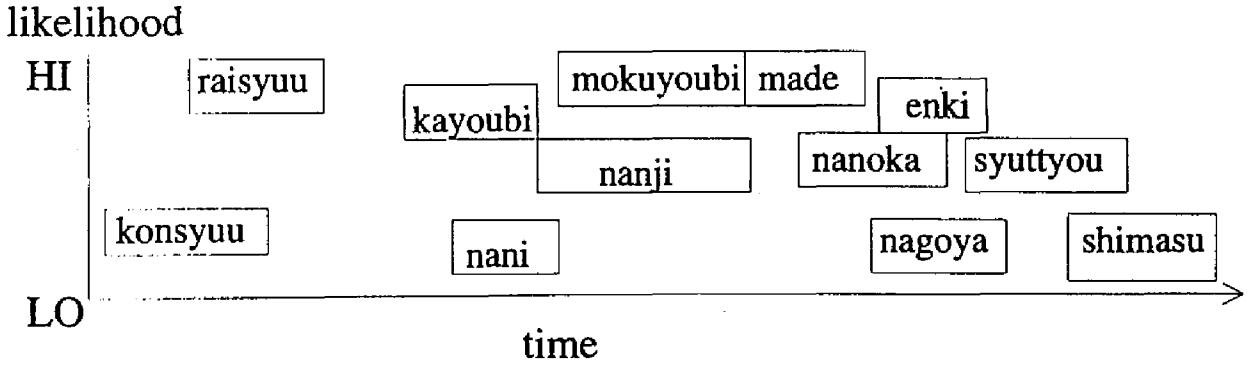


Figure 2.7: Sample word lattice

2.4.1 Path analysis

To start processing, the parser selects two keywords that have the highest score and do not overlap each other (slight overlapping is permitted). In this example, the keywords "*raisyuu*(next week)" and "*mokuyoobi*(Thursday)" are selected.

By tracing the network path of the two keywords, we get following three paths(see Figure 2.8), which are transformed to the semantic representation in Table 2.2. The snapshot after *raisyuu* and *mokuyoobi* picked up is shown in Figure 2.9.

(1) *raisyuu*(next week) → week → time → start_time
← time ← day ← *mokuyoobi*(Thursday)

(2) *raisyuu*(next week) → week → time → end_time
← time ← day ← *mokuyoobi*(Thursday)

(3) *raisyuu*(next week) → week → time → start_time
→ sentence ← end_time ← time ← day
← *mokuyoobi*(Thursday)

Table 2.2: Semantic representation obtained by *raisyuu* and *mokuyoobi*

path	semantic representation
(1)	[S,[start_time,[week,*raisyuu*],[day,*mokuyoobi*]]]
(2)	[S,[end_time,[week,*raisyuu*],[day,*mokuyoobi*]]]
(3)	[S,[start_time,[week,*raisyuu*]], [end_time,[day,*mokuyoobi*]]]

2.4.2 Verification of phrase

In case of (1) or (2), the path containing special relations "limit" , syntactic verification by phrase template is done. Some phrase templates are shown in Figure 2.10. Here, only verification of local modification of the words is done.

Phrase spotting for case (1) and (2) (e.g.,*raisyuu no mokuyoobi kara*(from Thursday next week)) failed, but this meaning hypothesis remains in treating spontaneous input.

The end of parsing is judged by semantic case check and the cover ratio of the lattice. If it does not reach the end of parsing, next keyword is picked up from the word lattice.

The next picked up keyword is "*made*(to)". The neighboring keyword, which is already picked up, is "*mokuyoobi*(Thursday)". (see Figure 2.11). Then, we get following two paths, which are transformed to the semantic representation in Table 2.3.

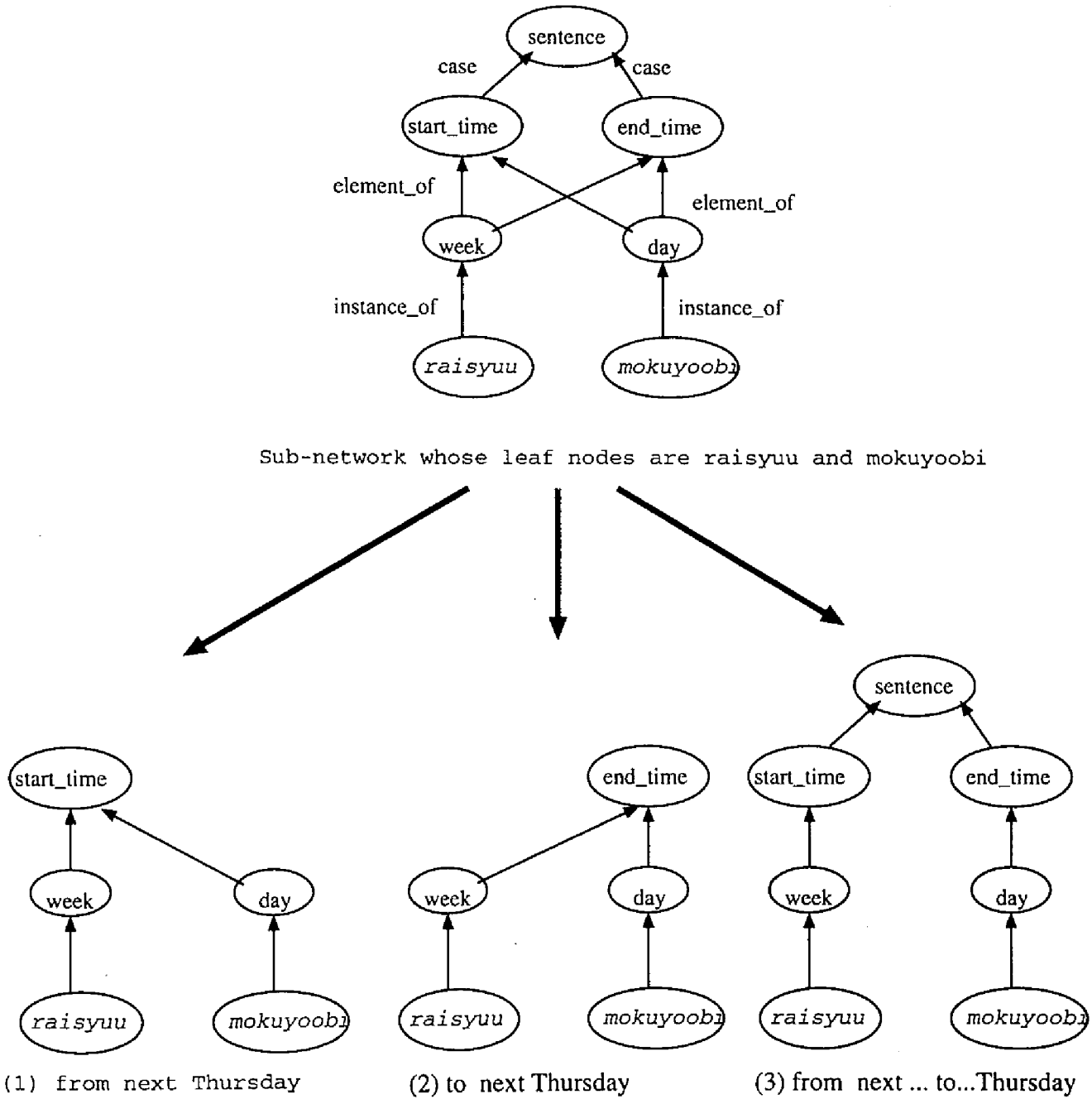


Figure 2.8: Getting network path

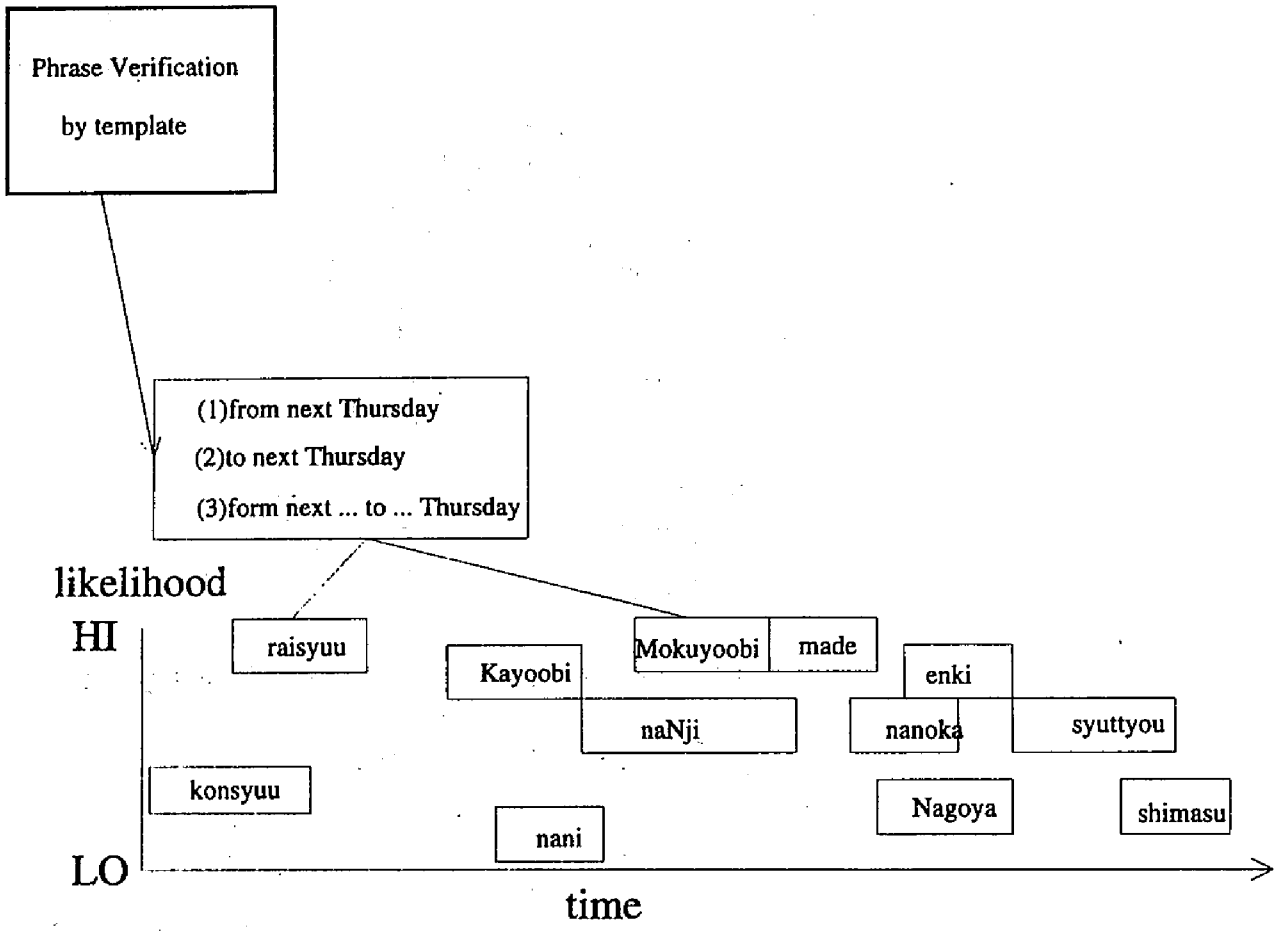


Figure 2.9: Keyword-driven parsing (1)

(a) *mokuyoobi*(Thursday) → day → time → start_time
→ sentence ← end_time ← index ← *made*(to)

(b) *mokuyoobi*(Thursday) → day → time → end_time
← index ← *made*(to)

```
p_template([day,p(conj),hour,pp_from],start_time).
p_template([neighbor,p(place)],place).
p_template([day,p(conj),attendance_event,p(obj)],event).
p_template([hour,p(conj),attendance_event,p(obj)],event).
```

Figure 2.10: Example of phrase template

Table 2.3: Semantic representation obtained by *mokuyoobi* and *made*

path	semantic representation
(a)	[S,[start_time,[day,*mokuyoobi*]],[end_time,_]]
(b)	[S,[end_time,[day,*mokuyoobi*]]

2.4.3 Construction of meaning hypothesis

Concatenation procedure is called to concatenate already constructed semantic representations ((1),(2),(3)) to new semantic representations ((a),(b)) (see Figure 2.12). Then, we get concatenated semantic representations listing in Table 2.4.

Table 2.4: Concatenated semantic representation

path	semantic representation
(1)-(b)	[S,[start_time,[week,*raisyuu*], [day,*mokuyoobi*]],[end_time,_]]
(2)-(a)	[S,[end_time,[week,*raisyuu*],[day,*mokuyoobi*]]]
(3)-(a)	[S,[start_time,[week,*raisyuu*]], [end_time,[day,*mokuyoobi*]]]

This concatenation is done according to the combination principle. This principle reflects a case principle like one case in a clause, contrast principle, complementary principle, etc. For example, [day,*mokuyoobi*] is the start time in candidate (1). On the other hand, the same word [day,*mokuyoobi*] is the end time in candidate (a). Thus, the combination of (1)-(a) is excluded. We get following combination of the meaning candidates.

We assume the next word is *kayoobi*(Tuesday). The neighboring keywords that are already picked up are *raisyuu* and *mokuyoobi*. The semantic representations are constructed by the paths form *raisyuu* to *kayoobi* and from *kayoobi* to *mokuyoobi*. Then, only one semantic representation, *raisyuu no kayoobi kara ... mokuyoobi made ...*, that is combined

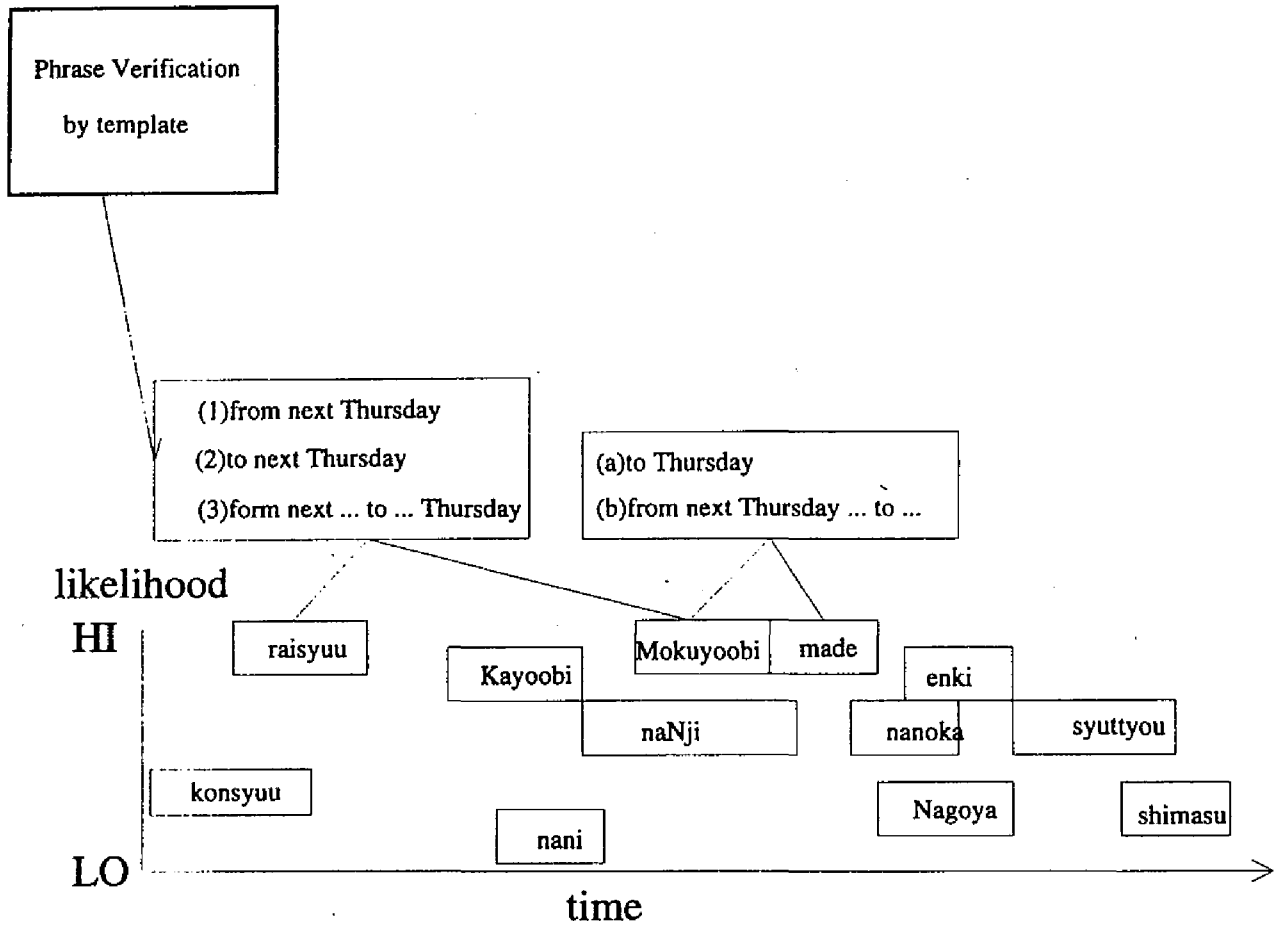


Figure 2.11: Keyword-driven parsing (2)

with (3)-(a), remains (see Figure 2.13).

After that, Skipping invalid words by the word order constraints or semantic constraints, the word *shuttyou*(business trip) and *Nagoya* is included into semantic representation (see Figure 2.14).

The final semantic representation of this utterance is following:

```
[assert,[start_time,[week,raisyuu],[day,kayoobi]],
  [end_time,[day,mokuyoobi]],
  [event,[event,shuttyou]],[place,Nagoya]].
```

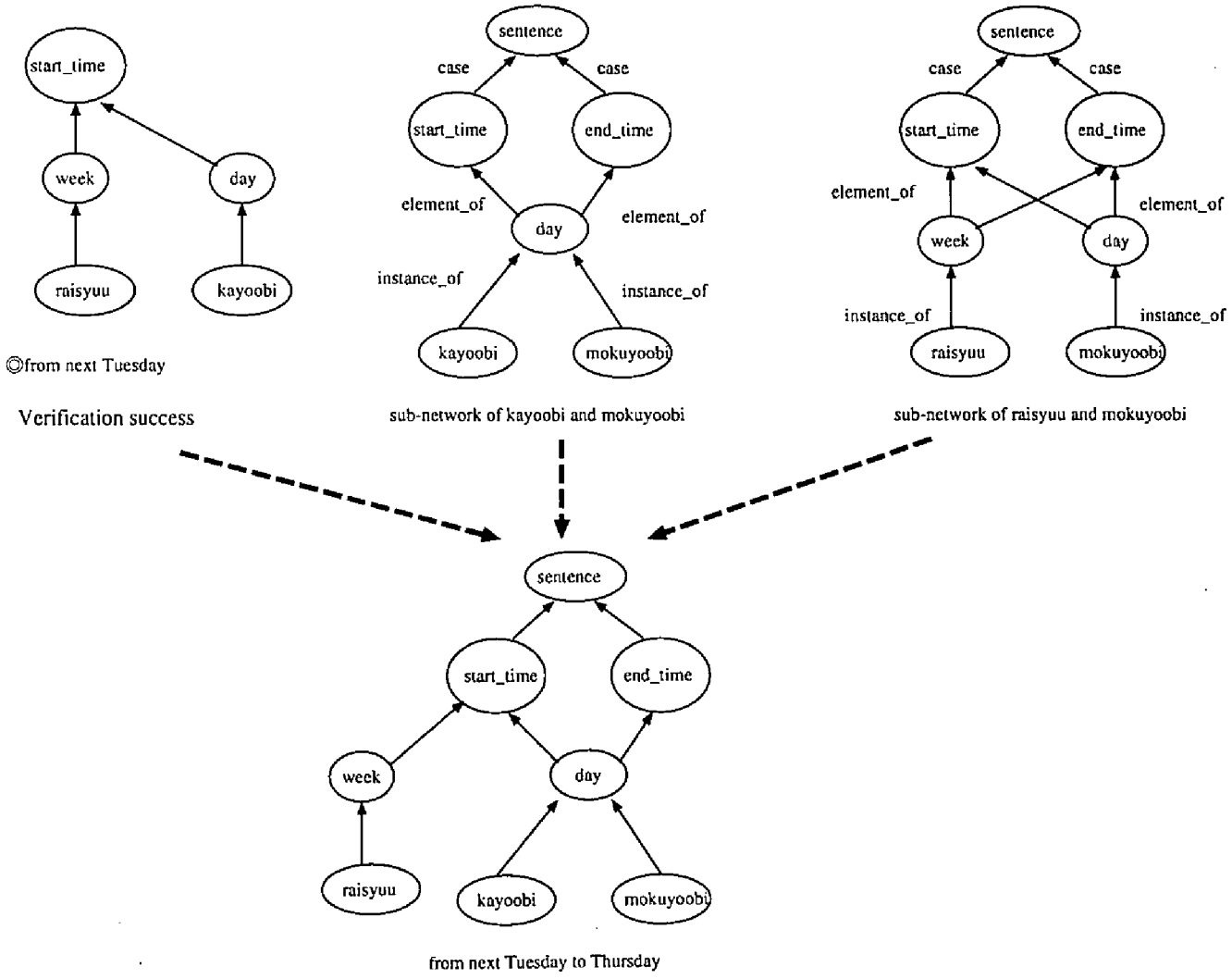


Figure 2.12: Constructing new semantic representation

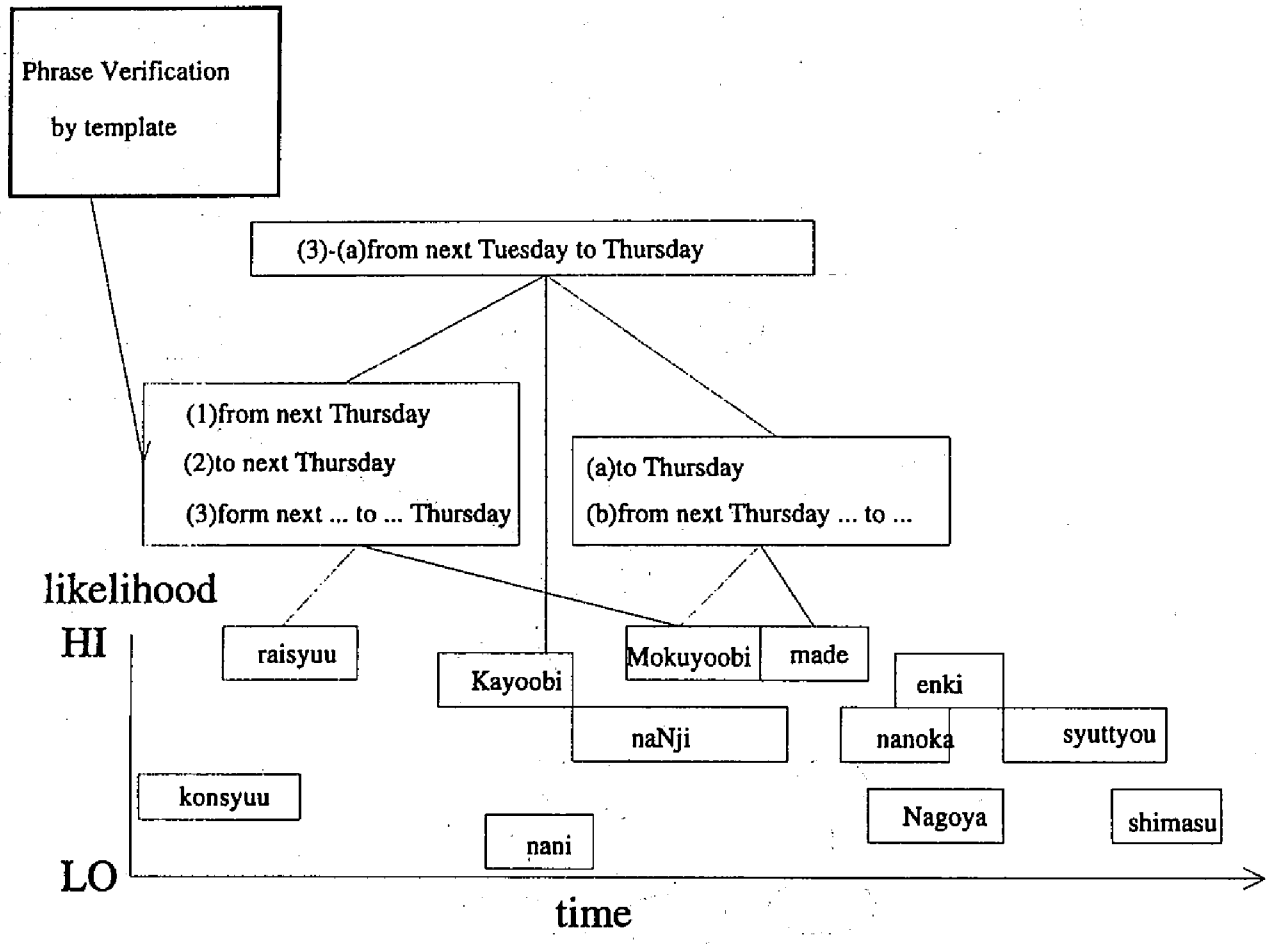


Figure 2.13: Keyword-driven parsing (3)

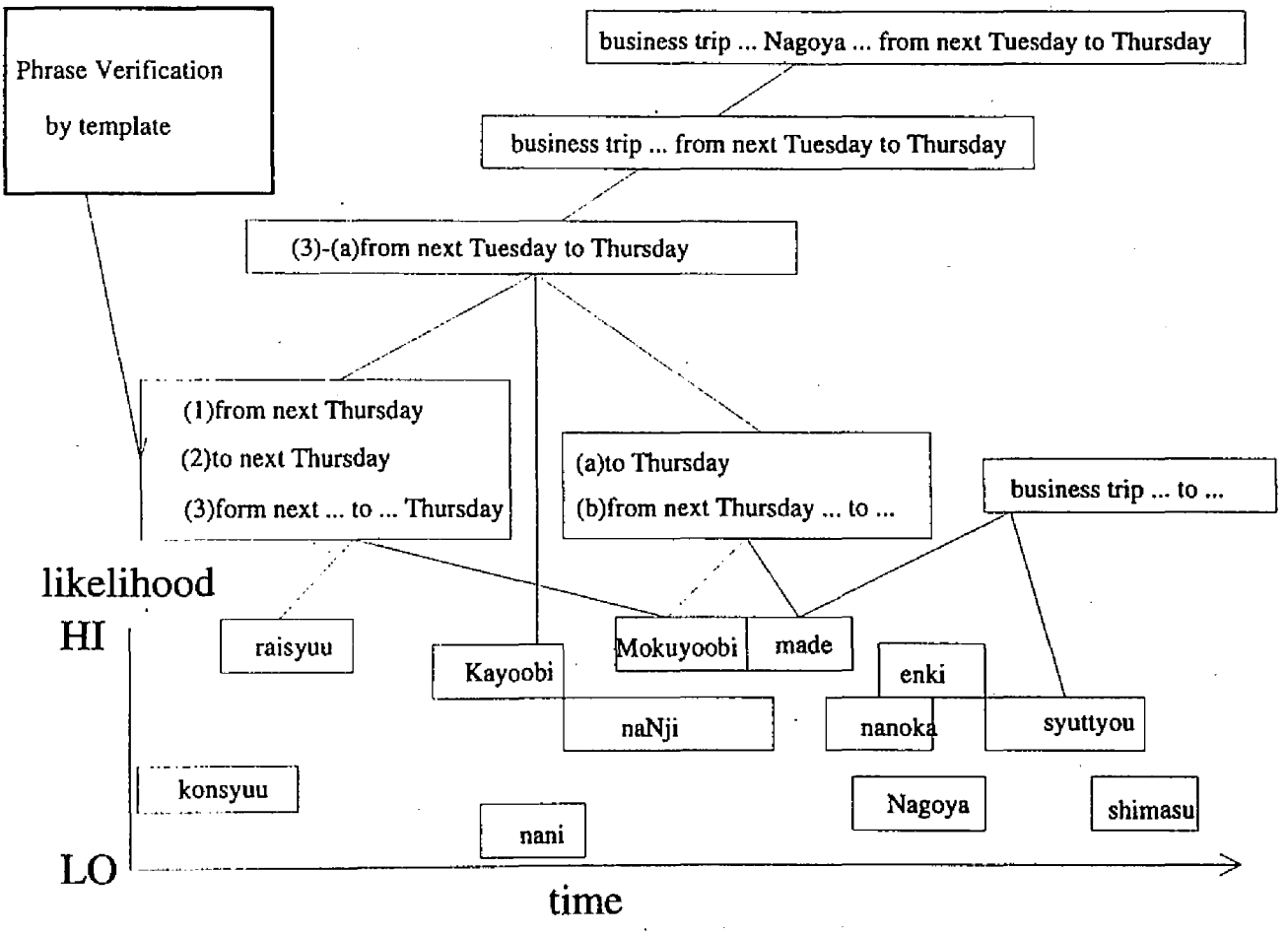


Figure 2.14: Keyword-driven parsing (4)

2.5 Experimental results

We made an experiment to evaluate the performance of keyword-driven parser. We used 75 kinds of sample Japanese sentences each of which was uttered by 8 male speakers. In these sentences, 50 sentences are grammatical and 25 sentences are spontaneous, which include filled pauses and inner phrase pauses. These spontaneous speech data are read by the speakers, but the phenomena which are included the speech are taken from our original task corpus. The list of grammatical sentences are shown in Appendix I, and spontaneous sentences are shown in Appendix II. The task of these test sentences are personal scheduling, which include assert, modify, delete, query sentences. The vocabulary size is 236 words, and the number of phoneme per word is 7.47 phoneme.

2.5.1 Parsing approach

If the spoken dialogue system assumes the input speech as a grammatical sentence, the best way of offering the constraints to acoustic analyzer or analyzing the sentence to translate semantic representation is using grammar. If it is written in context free grammar formalism, it can use the efficient parsing algorithm. But the defect of using such grammar is weakness to the ungrammatical sentences. We examine the performance of parsing approach considering these features.

Using probabilistic grammar

We calculate the frequency of the use of grammar rules, and add a probability to each grammar rule. Table 4.7 shows the result of recognition. Semantic accuracy is the ratio of the samples to which the parser outputs correct semantic representation. Dialogue continuation rate is the ratio of the samples to which only one value of slot of semantic representation (except for verb slot) is wrong, which can be corrected by the dialogue.

Table 2.5: Speech recognition results with PCFG(%)

	Semantic accuracy	Dialogue continuation rate
CFG	60.8	80.5
PCFG	65.5	84.8

Using dialogue constraints

We also examine the effect of dialogue level constraints. Because of the easiness of situation setting, it is evaluated in automaton dialogue model shown in Figure 1.6. Table 4.8 shows the result of recognition.

Table 2.6: Speech recognition results with dialogue constraints(%)

	Semantic accuracy	Dialogue continuation rate
CFG	60.8	80.5
CFG+Dialogue	65.3	83.8

2.5.2 Spotting approach

The ill-formedness of spontaneous speech is hard to model and describe all of them. Strict parsing approach fails when it meets such ill-formedness. Then, we need spotting approach which skips unanalyzable part of utterance. We examine the performance of word spotter and semantic analyzer. Table 2.7 shows the result of recognition.

Table 2.7: Speech recognition results in word spotting approach(%)

	Semantic accuracy	Dialogue continuation rate
Grammatical(A*)	64.0	81.5
Spontaneous(A*)	45.0	65.0
Grammatical(spot)	44.0	59.5
Spontaneous(spot)	27.5	41.0

2.5.3 Comparison of two approaches

In comparing these approaches, parsing approach shows higher recognition rate in grammatical input. Also, in spontaneous speech, parsing approach wins spotting approach because of (1) grammatical constraints still works in slight deviation of correct sentences, and (2) the performance of word spatter is very low which compensate for ill-formedness of input utterance. Therefore, we conclude that if we aim to deal with slight ill-formedness,

it is enough to use parsing approach. But we aim to deal with real spontaneous speech, we must make more precise word spotter.

2.6 Discussion

We intend further improvement at two points. (1) Current implementation of verification of phrase is limited to word lattice. Verification with spotting function words will increase phrase recognition rate. (2) Current scoring method is so simple (the sum of keyword's score) that unrecognized segments are not evaluated. It needs another scoring method that can evaluate overall input.

2.7 Summary

we presented a keyword-driven speech parser as a robust language processing. In our approach, the seeds of semantic analysis are words in the same way as task-oriented approach. But partially using grammar, the relationship of the words are drawn from general semantic network description. We call it 'semantic-based approach.' Dialogue level predictions can be used in our method by limiting the search space in activated subnetwork. By this method, we realize semantic analyzer that achieve 69.3% in semantic understanding rate, and 87.5% dialogue continuation rate (less than one error contained in keyword, except for verb).

Chapter 3

Cognitive Process Model of Cooperative Spoken Dialogue

3.1 Introduction

In this chapter, we propose a cognitive process model of spoken dialogue. In order to make an interactive dialogue system, we need two management processes: one is an understanding process which manages the subprocess from utterance understanding to response generation; the other is a dialogue management process which aggregates the utterances into the discourse segment, and manages focus and intentions of dialogue. Furthermore, in applying the dialogue model to spoken dialogue systems, we have to deal with input errors caused by speech recognition errors. In previous researches of spoken dialogue, these three aspects are treated independently.

In the research of whole processing mechanism, that is, from utterance understanding to response generation, there are two major approaches: one is parallel multi agent with distributed databases [18], the other is sequential processing combining some module [19], [20].

From the viewpoint of the usage of various constraints and cognitive modularized mechanism, multi agent approaches are good model of dialogue processing of human being. However, constraints satisfaction problem of multi agent is difficult to implement, and hard to control. Then, sequential processing is widely hired in implementing dialogue systems.

In order to implement intelligent dialogue system by sequential approach, they must have dialogue management unit which can be accessed from processing modules at any time. Grosz et. al. proposed dialogue management mechanism which has three elements:

linguistic structure, intentional structure and attentional state [21]. In [21], they refereed the phenomena that the meaning of utterance makes discourse context, and inversely, the discourse context restricts the possible meaning of utterance. But they didn't mention the relation between these three elements and utterance understanding mechanism.

In order to grasp linguistic structure, there are two major management method: stack structure ([21], [22], [19]) and AND-OR tree structure ([23], [24], [25]). Stack structure is easy to implement and has simple relation to the attentional state. However, it is hard to manage the movement to subdialogue, to realize variable initiative, and to make collaborative response form the task level. In addition, contextual information which is popped from the stuck cannot be accessed in principle. Such contextual information may be useful in repairing misunderstandings caused by speech recognition errors. On the other hand, because AND-OR tree structure is a kind of representation of task structure, dialogue management by AND-OR tree confuses linguistic structure and intentional structure. Therefore, deviated subdialogue from problem structure, e. g. clarification dialogue, meta dialogue about system's ability etc., should treat independent way of main task structure. As a consequence, the dialogue management suitable for spoken dialogue needs accessibility to previous context and should distinguish linguistic structure with intentional structure. In [20], Airenti et. al. proposed an cognitive process model which has two information structure: dialogue game and behavior game. These correspond to linguistic structure and intentional structure respectively. However, these information structure did not be specified enough to use dialogue systems.

from surface understanding to response generation, which describes the transformation of dialogue participant's beliefs [20]. In their model, they divided the cognitive process in five steps: (1) literal meaning, where the reconstruction of the mental states literally expressed by the actor takes place; (2) speaker's meaning, where the partner reconstructs the communicative intentions of the actor; (3) communicative effect, where the partner possibly modifies his own beliefs and intentions; (4) reaction, where the intentions for the generation of the response are produced; and (5) response, where an overt response is constructed. They represent the knowledge of dialogue management as two kinds of games; conversational game and behavioral game. Their model is constructed for explaining the cognitive process in one turn, that is from utterance understanding to response generation of one dialogue participant. The dialogue management part and its interaction between cognitive process are not specified enough to implement dialogue

systems.

Furthermore, some kind of error correction mechanism is inevitable for the dialogue model for spoken dialogue systems. In treating the speech recognition errors in previous researches, the main point was put into implementing robust parsing [26]. For the most part of dialogue research, the input of the model is the correct semantic representation of the user's utterance. However, major limit of robust parser is the phenomena of replacing a word by the same syntactic / semantic categorical word, e.g. if *Monday* is replaced by *Sunday*, robust parser cannot find out the replacement by its syntactic / semantic knowledge. In addition, the lack of selectional case word(s) cannot be found out by robust parser. Therefore, we have to give consideration the speech recognition error management in dialogue model.

From the above discussion, we decided that the major points in constructing dialogue model are closely combining sequential module, distinguishing linguistic structure and intentional structure, and constructing robust dialogue manager. Our dialogue model is a cognitive process model (1) which integrates the specified phased processing from utterance understanding to response generation, (2) which specifies the interactions between the processing of each steps and dialogue management mechanism, and (3) which identifies the possible errors caused by recognition error and the method of recovering the error. By implementing this model to the dialogue system, we expect to realize a robust and cooperative spoken dialogue system.

The remainder of this chapter is organized as follows. 3.2 is devoted to survey previous works in dialogue processing. Cognitive process of proposed model is described in 3.3. The management mechanism of linguistic structure is explained in 3.4 and management mechanism of intentional structure is explained in 3.5. The example of dialogue processing is described in 3.6. Conclusions are given in 3.7.

3.2 Survey of dialogue processing

The dialogue system interacts with human in several domains, such as, question-answering systems, reasoning systems, planning-assistant systems, etc. Such systems are viewed as conversational agent that has a specific memory system and specific processing mechanism suitable for conversation. In this section, we survey previous dialogue processing researches by explaining the requirement to implement such conversational agent from

general point of view.

In order to build an intelligent conversational agent, Allen enumerated the elements of the agent architecture [27]. These elements are : perceiving, beliefs, desires, planning, commitment, intentions and acting. Among these elements, beliefs, desires and intentions are part of the agent's cognitive state. The other elements, perceiving, planning, commitment and acting are processing. The relations of each elements are shown in Figure 3.1. This model is called BDI (beliefs, desires and intentions) model [28] of conversational agent.

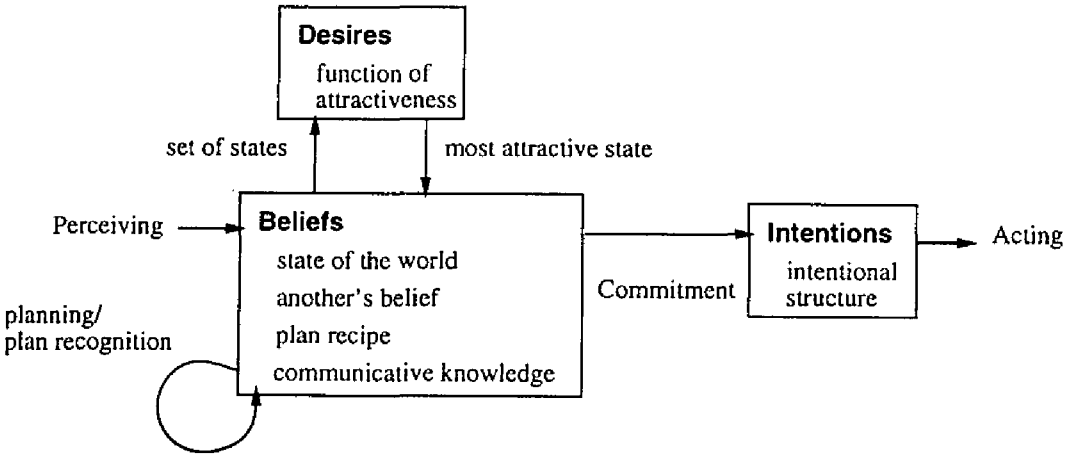


Figure 3.1: BDI model of conversational agent

The major point of previous works of dialogue processing can be located in this BDI model. Following subsections are dedicated to the survey of dialogue processing along with the elements of the BDI model.

3.2.1 Perceiving the language

Considering the language as a kind of rational act, it has a hierarchy depending on the contextual condition. Concerning the hierarchy, Austin [29] observed that there are three acts performed whenever something is said:

1. locutionary act

   the act of uttering a sequence of words,

2. illocutionary act

the act that the speaker performs in saying the words,
ex) inform, request, promise, ...

3. perlocutionary act

the act that actually occurs as a result of the utterance.
ex) convince, motivate, ...

Also, Searle [30] examined the contextual condition as *felicity condition*, which enables to generate the higher level act from the lower level act. For example, locutionary act of saying "See you at 8 p.m. tonight" can be interpreted as illocutionary act of promise when felicity conditions, such as the hearer can understand English sentence, speaker uttered this sentence before 8 p.m. at that day, etc.

In the context of isolated sentence understanding (described in chapter 3), perceiving the language means to extract meaning representation from the sentence. On the other hand, in the context of dialogue processing, the major point of perceiving the language is to identify illocutionary act and to presume the perlocutionary act which the speaker intended. The required knowledge in perceiving the language is represented in agent's belief space as a meta belief.

3.2.2 Beliefs of conversational agent

Beliefs of cognitive states and another's belief

What kind of beliefs are needed in conversational agent? Because conversational agent is a kind of rational agent, they need representations of their own cognitive states (their own mental state and the representations of the world). Such mental state can be represented as beliefs using a predicate, e.g.

$$\mathrm{BEL}_{system}p$$

System stands for a conversational agent. P stands for the Proposition.
By using belief, knowledge can be defined as:

$$\mathrm{KNOW}_{user}p \equiv p \wedge \mathrm{BEL}_{user}p$$

User stands for a conversational agent. This definition means that user *knows* a proposition p iff p is true and user believes p. It seems problematic that a proposition p appears directly. However, there is no problem in practice because KNOW operator is used in other agents beliefs, e. g. $\mathrm{BEL}_{system}\mathrm{KNOW}_{user}p$.

Moreover, in order to communicate with other agent, they need beliefs of another agent's belief. Such beliefs can be represented by nesting *Bel* predicate, e.g.

$$\mathrm{BEL}_{system}\mathrm{BEL}_{user}p$$

However, such expression cannot represent the thing that *user* knows but *system* does not know. For example, the precondition in asking the current time (*system* knows that *user* knows what time it is.) cannot be represented because it contains uninstantiated variable: current time.

In case the proposition contains uninstantiated variable, we need two operators: KNOWREF (agent knows the value) and KNOWIF (agent knows the truth of the proposition), which are defined as follows:

$$\mathrm{KNOWREF}_{system}\lambda x p_x \equiv \exists x \mathrm{BEL}_{system}p_x,$$

$$\mathrm{KNOWIF}_{system}p \equiv \mathrm{BEL}_{system}p \vee \mathrm{BEL}_{system}\neg p.$$

But how deep is a nesting of beliefs required to understand dialogue? One can easily construct an example which needs any nesting levels. Then we need the representation of shared belief defined below:

$$\mathrm{SH}_{system,user}p \equiv \mathrm{BEL}_{system}(p \wedge \mathrm{SH}_{user,system}p).$$

(That is, $\mathrm{SH}_{system,user}$
$\supset \{\mathrm{BEL}_{system}p,\ \mathrm{BEL}_{system}\mathrm{BEL}_{user}p,\ \mathrm{BEL}_{system}\mathrm{BEL}_{user}\mathrm{BEL}_{system}p,\ \ldots\}$.)

Notations of action and intention

In order to represent communicative knowledge using, we define the notations of action and intention here. Action type is represented by a the operator DO, which is used following way:

$$\mathrm{DO}_{user}\text{lit-illoc}(system, p, f).$$

(User performs literal illocutionary act with system, propositional content p and illocutionary force f.)

On the other hand, the action type also can be represented by its effect using the operator DONE

$$\mathrm{DONE}_{user}\text{closed(Window)}$$

Intention is represented in the same form of an action type:

$$\mathrm{INT}_{user}\text{lit-illoc}((system, p, f),\ \mathrm{INT}_{user}\text{closed(Window)}$$

Beliefs of communicative knowledge

In order for conversational agent to communicate each other, a general scheme of communicative act should be hold as $SharedBelif$. For modeling dialogue participant's belief, it is simple to treat these communicative knowledge as meta belief. The scheme of communicative act provides the connection of three levels of speech act; locutionary act, illocutionary act, and perlocutionary act. For example, figure 3.2 shows the scheme of a communicative act $MotivateByRequest$, which connects *Request* to perlocutionary act *Intend*, and a decomposition rule which connects literal illocutionary act *Imperative* to illocutionary act *Request*.

Furthermore, in order to maintain cooperative dialogue, the agent should have the ability of planning their own action and recognizing another agent's plan. Therefore, conversational agent needs to have plan recipe in their beliefs. The plan recipe describes the sequence of actions and their generation relations for the purpose of the specific goal. For example, the plan recipe of *Register_meeting* by using the same description of the communicative act is shown in Figure 3.3. The leaf level of decomposition is connected o the communicative act. Such script-like representation of plan is following the work of Schank [31] and Allen [27].

Such a representation is useful both in planning and plan recognition if the user proceeds the dialogue following the script. But when user skips a certain step, say, the user

The Action Class *MotivateByRequest(e)*:

Constraints:	$Action(Act)$
Effects:	$\text{INT}_{system} Act$
Decomposition:	$Request(User, System, Act)$
	$DecideTo(System, Act)$

The Illocutionary Act *Request(e)*:

Decomposition1:	$Imper(User, System, Act)$
Decomposition2:	$Interrog(User, System, CanDo(System, Act))$

Act: any action expression.

Figure 3.2: Representation of communicative act and illocutionary act

The discourse script *Register_meeting(e)*:

Roles:	$System, User, Event, Time, Place$
Decomposition:	$Greetings(System, User)$
	$GetEventInfo(System, User, Event)$
	$GetTimeInfo(System, User, Time)$
	$GetPlaceInfo(System, User, Place)$
	$VerifyContents(System, User, Event, Time, Place)$
	$Closing(System, User)$

The discourse script *GetEventInfo(e)*:

Roles:	$System, User, Event$
Decomposition:	$MotivateInformRef(System, User, \lambda x Event = x)$
	$ConvinceByInformRef(User, System, \lambda x Event = x)$

Figure 3.3: Representation of plan recipe

refers the place before telling the meeting time, the system cannot follow the dialogue. In order to follow the variety of dialogue strategies, the system must have plan recipe as much as the variety of dialogue.

The hierarchical representation of plan and actions is one solution of above mentioned problem. The most representative work is Kautz's Event Hierarchy [32] (see Figure 3.4). It represents the decomposition relationship between plan and actions and abstraction relationship between plan and subplans. In recognizing the partner's plan, it uses minimal cover that includes all the actions previously observed. In planning the actor's action, it refers the ordering constraints which are attached to each decomposition link if necessary.

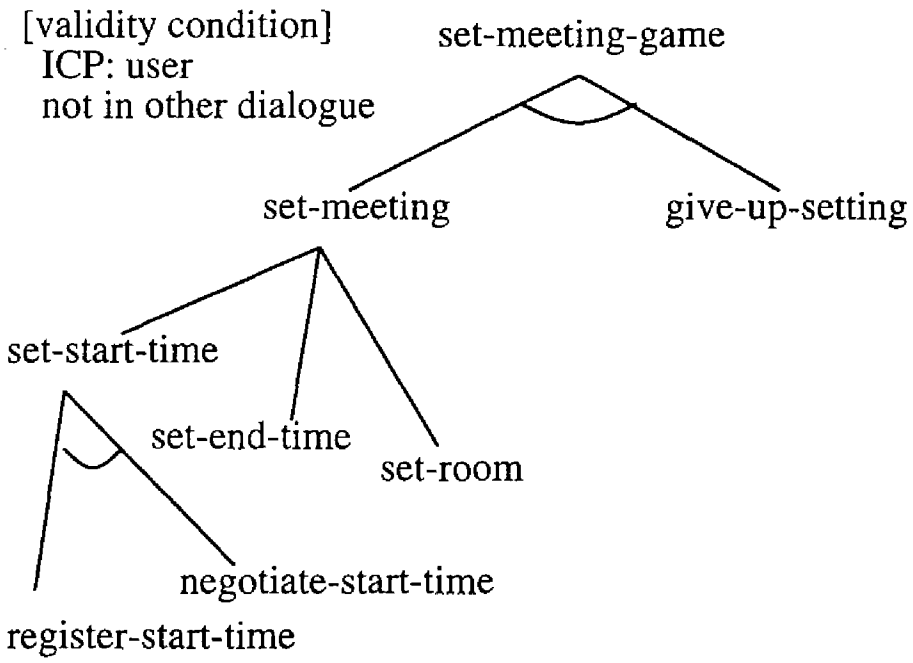


Figure 3.4: Set-meeting-game

3.2.3 Desire as a criteria of the attractiveness of the state

There are little works about the desire of conversational agent. Allen's formalization of the desire as a criteria of the attractiveness of the state is a concise modeling [27]. For example, in scheduling system, the state where there is nothing to do is the most attractive state. If the system is asked to register a meeting by the user, the system transits the state where it has a problem to be solved, i.e. less attractive state. Then system tries to transit the former state by solving the problem. Therefore, in task-oriented dialogue

between human and computer, we can define system's desire as a meta rule such that the system tries to solve problem in order to get back the normal state.

In case the system have to give up to solve all the subproblems given by the user, normal planning procedure cannot decide which problems should remain unsolved. The dialogue system which has to deal with such conflicting situation should have some criteria for measuring the attractiveness of the any state.

3.2.4 Planning and commitment

In the context of BDI model, planning means the generation of (partial) plan recipe that can lead the agent to the more attractive state. Following this formalization, commitment is the process to choose the (partial) plan recipe. The plan recipe is selected under several constraints and preference conditions (e.g. attentiveness constraint, sincerity condition, sincerity preference, shared knowledge preference, helpfulness preference, conciseness preference, etc.) As a result of commitment procedure, the system comes to have plan as a goal, we call it *future-directed intention*, which is almost equal to the selected plan recipe that have been hold until it is fulfilled or abandoned.

In [33], Biermann et al. proposed a planning and commitment mechanism in dialogue by Prolog like AND-OR tree (Figure 3.5) which represents the structure of the problem.

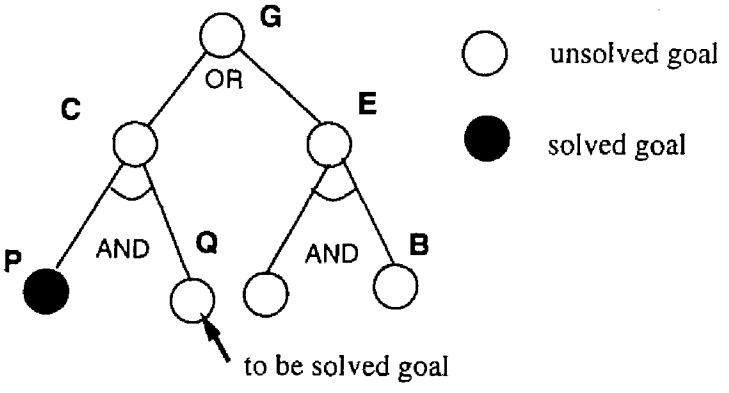


Figure 3.5: AND-OR tree of problem structure

The planning follows the behavior of Prolog program. The system tries to make leaf node true in depth first order. In addition, the system have a variable initiative from directive to passive, which varies the system commitment. Two examples of the dialogue are shown in Figure 3.6.

- keep initiative

  S: Is the switch up?
  U: B is true.
  S: Yes, but is the switch up?

- release initiative

  S: Is the switch up?
  U: B is true.
  S: I see, then ...

Figure 3.6: Example dialogue of variable initiative

3.2.5 Intentions in communication

There are two notions of intention. One is *intention in action*, and the other is *future-directed intention*. In the context of dialogue research, the later is mainly discussed.

One definition of intention is, contrary to our model, a input of planning process. In Cohen and Levesque [34], intention is defined by using persistent goal of the agent:

$$(INTEND_1\ user\ e) \stackrel{\text{def}}{=} (P-GOAL\ user[DONE\ user, (BEL\ user(HAPPENS\ e))?; e])$$

e: any action expression, ?: operator for evaluating if the proposition is true.

But the planning process itself was not formalized in their paper.

The other definition of intention, the same treatment to our model, is a output of the planning process. In other words, such a intention is derived from the deliberation in one's belief space by process of planning and commitment based on the communicative knowledge shown in Figure 3.2.

Such definition of intention is compatible to the planning and plan recognition procedure. However, in such treatment, the discrepancy of plan recipe between dialogue participants cannot be dealt with. For treating such difference in knowledge, Pollack represented the plan as complex mental attitudes [35]. In Pollack's theory, the plan recipe is represented by the form of *SimplePlan* (see Figure 3.7).

This *SimplePlan* is constructed by the beliefs (*user* believes the actions can be executable and the generation relation between actions is valid) and intentions corresponding

$$SIMPLE\text{-}PLAN(user, \alpha_n, [\alpha_1, \ldots, \alpha_{n-1}], t_2, t_1) \leftrightarrow$$

1. $BEL(user, EXEC(\alpha_i, A, t_2), t_1)$,for i=1,...,n $\wedge$
2. $BEL(user, GEN(\alpha_i, \alpha_{i+1}, A, t_2), t_1)$,for i=1,...,n$-1$ $\wedge$
3. $INT(user, \alpha_i, t_2, t_1)$,for i=1,...,n $\wedge$
4. $INT(user, by(\alpha_i, \alpha_{i+1}), t_2, t_1)$,for i=1,...,n$-1$

Figure 3.7: Representation of SimplePlan

to each beliefs.

Grosz and Sidner extended the *SimplePlan* to the planning process of multi agent *SharedPlan*, by replacing BEL to SH (in their term, Mutual Belief) [36].

In these frameworks, intention is located in part of mental state that construct agent's plan.

3.2.6 Acting

Once dialogue system has an intention to react to the user's utterance, and also there is no trouble in reacting, system generates response as an acting process. However, there exist some trouble in simply reacting, the system should select its action.

In [37], van Beek and Cohen described the response generation mechanism in case of user's plan is ambiguous. The algorithm is shown in Figure 3.8.

1. Getting plan hypotheses by plan recognition
2. Add label to each hypothesis by defect evaluation. The labels are (1)failure of presupposes (2) order error (3) existing better plan (4) no defect.
3. Make clarification dialogue until all the labels are same.
4. Make response by user's query and plan hypotheses.
   (a) if all the plans are no defect, make simple response.
   (b) if all the plans have the same label, make the response which points out the defect. (ex. failure of presupposes)

Figure 3.8: Response generation Algorithm

3.2.7 Dialogue model as cognitive process modeling

In previous subsection, we overviewed the previous works about dialogue processing form the modularized functional point of view. But the practical dialogue processing does not always proceed following the informational flow of the modularized functional model (BDI model). When hearer cannot understand the word or utterance, they probably respond the word "I beg your pardon?" without modifying his/her beliefs. Such a control of dialogue processing can be understood from the viewpoint of cognitive process.

Airenti et al proposed a dialogue model as cognitive process model (CPM) [20]. They intended to analyze the process of comprehension of a communicative act in five steps: (1) literal meaning, (2) speaker's meaning, (3) communicative effect, (4) reaction, and (5) response. Also, they used a notion of two kinds of game, conversational game and behavioral game. These games reflect the distinction of communicative cooperation which must be maintained through communication, and behavioral cooperation which is not necessarily maintained by the partner. Although this model was not designed for intending to deal with spoken dialogue, our dialogue model is largely influenced this model. Therefor, we explain this model in detail here.

Five steps cognitive process model

CPM distinguished the process of comprehension of a communicative act in following five steps.

1. Literal meaning ... mental state expressed by A is reconstructed from the literal illocutionary act
2. Speaker's meaning ... B reconstructs the A's communicative intentions [1]
3. Communicative effect
   (a) attribution ... B attributes to A private mental states like beliefs and intentions

[1] Communicative intention is an extended notion of infinite nesting of intention, that is, intention to achieve an effect on the partner and the intention that the previous intention to be recognized. When the user has an intention that the two following facts be shared by the system and the user: that proposition p, and the user intends to communicate p to the system:

$$\mathrm{CINT}_{user,system}p \equiv \mathrm{INT}_x\mathrm{SH}_{system,user}(p \wedge \mathrm{CINT}_{user,system}p).$$

(that is, $\mathrm{CINT}_{user,system}p \supset \{\mathrm{INT}_{user}\mathrm{SH}_{system,user}p,\ \mathrm{INT}_{user}\mathrm{SH}_{system,user}\mathrm{INT}_{user}\mathrm{SH}_{system,user}p, \cdots\}$.)

(b) adjustment ... B's mental states about the domain of discourse are possibly modified as a consequence of A's utterance

4. Reaction ... the intention for generating the response are produced

5. Response ... an overt response is constructed

Each step is controlled by conversational game. Behavioral game is used in the inference of each step. The overall notion of this five steps model is illustrated in Figure 3.9.

Conversational game and behavioral game

In CPM, two kinds of game are introduced in describing the process of communication, that is conversational game and behavioral game.

Conversational game is a set of meta rules which control the overall process. The task of each step is described by the notion explained in previous subsection. For example, the conversational game in literal meaning step is shown in Figure 3.10.

On the other hand, behavioral game is a behavior plan which is shared between dialogue participants. The example of behavioral game is shown in Figure 3.11.

Problems of CPM

There exist some problems in applying CPM to spoken dialogue systems.

First, CPM does not specify the interaction between five step processing and dialogue management mechanism. In order to apply CPM to actual dialogue systems, we have to specify the mechanism of two kinds of dialogue management: conversational management and problem solving. In addition, we have to define the interaction between five step processing and such dialogue management mechanism.

Second, CPM does not assume errors which can occur in some level in spoken dialogue systems. For example CPM assumes that there exists no problem in understanding literal meaning. But in spoken dialogue systems, there are various ambiguity and uncertainty of user's input, such as uncertainty of speech recognition results, syntactic and semantic ambiguity, ill-formed utterances and uncertainty of user's intention.

Considering above problems, we intend to design a cognitive process model for spoken dialogue which involves dialogue management mechanism and error recovery strategy /

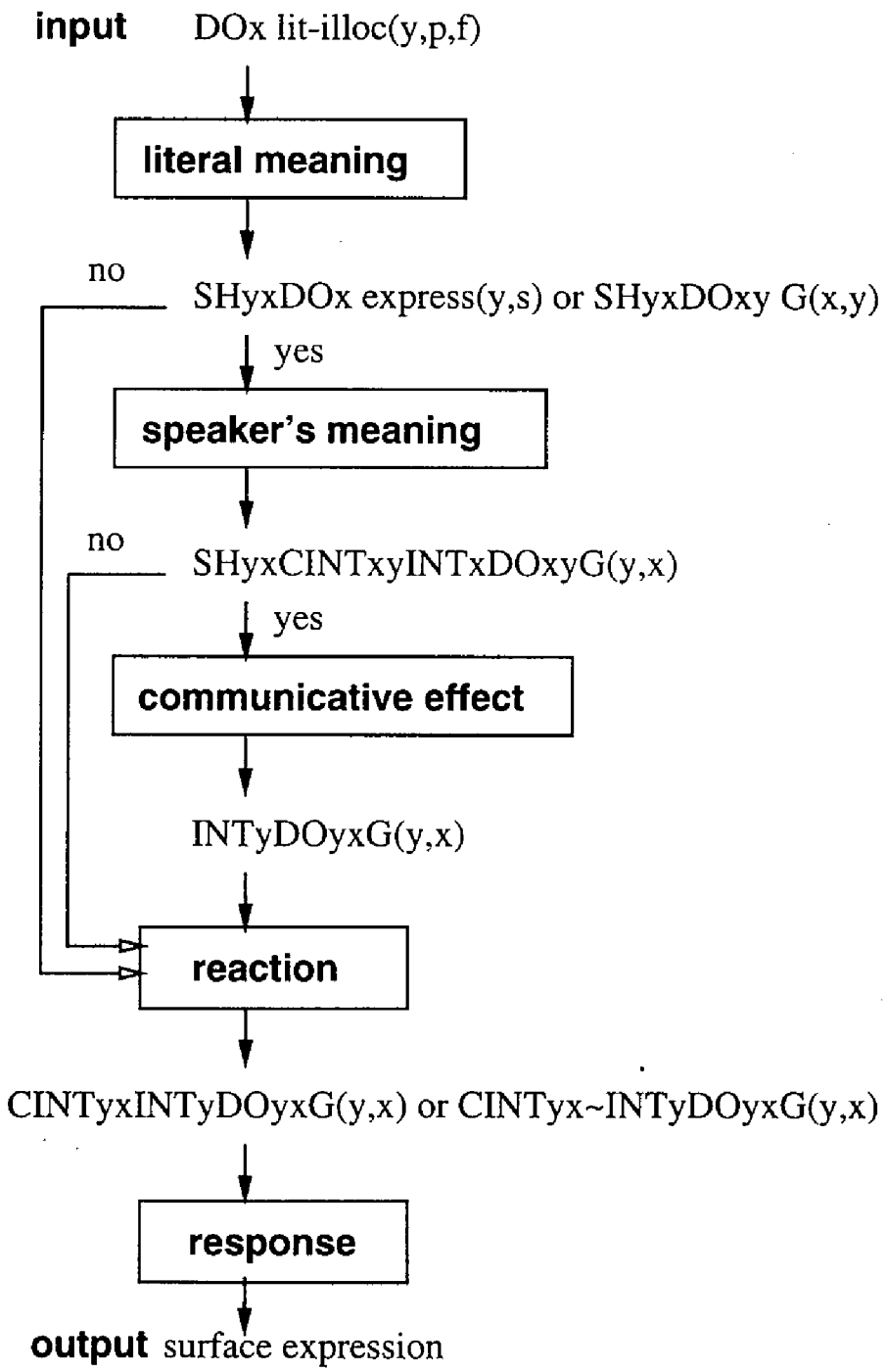


Figure 3.9: Airenti et al's cognitive process model

Metarule M1:
task: $SH_{yx}DO_x express(y,s) \vee SH_{yx}DO_{xy}G(x,y)$
if fulfilled: activate understanding of speaker's meaning
otherwise: activate reaction

Figure 3.10: Example of conversational game

```
[KITCHEN]
validity condition: at home, after meal
- x does the dishes
- y makes something useful.
```

Figure 3.11: Example of behavioral game

technique in several processing level.

3.3 Cognitive process model of dialogue

In this section, we propose a cognitive process model for cooperative spoken dialogue systems (CPM-SDS). Basic phases of this model follows CPM but it is improved for applying spoken dialogue systems following the line of discussion in previous survey of CPM. The our CPM-SDS has generality for the treatment of communication errors because recovering strategies are taken into cognitive process instead of into semantic analysis or pragmatic problem solving stages.

In CPM-SDS, we use almost same operator with CPM. Although the knowledge structure of CPM is propositional level, we need first-order predicate level of knowledge structure that treat practical dialogue. The strong point of first-order predicate level is introducing KNOWREF operator, which can represent the other dialogue participant's knowledge without specifying all the parameter.

The operators used in CPM-SDS are listed in Table 3.1.

We have specified the utterance understanding and generation mechanism for spoken dialogue systems based on Airenti's cognitive process model. We redefine the steps as (1) meaning understanding, (2) intention understanding, (3) communicative effect, (4) reaction generation, and (5) response generation (see Figure 3.12). Also, we specified the interaction between cognitive process and dialogue management subsystems. By these extensions, the model can deal with errors which occur at each steps in processing.

Figure 3.13 shows the information flow of overall processing and processing in each

Table 3.1: Operators used in CPM-SDS

operator	definition	usage
bel(X, P)	primitive	X believes P
know(X, P)	P ∧ bel(X, P)	X knows P
knowif(X, P)	(P ∧ bel(X, P)) ∨ (¬ P ∧ bel(X, ¬ P))	X knows whether P is true or false
knowref(X, P)	∃ Z. bel(X, P(Z))	X knows Z which satisfy P(Z)
shared_bel(X, Y, P)	bel(X, (P ∧ shared_bel(Y, X, P)))	X and Y mutually believe P
do(X, E)	(primitive)	X does E
do(X, Y, G)	Primitive	X and Y has goal G
int(X, E)	(primitive)	X intends E
cint(X, Y, P)	int(X, shared_bel(Y, X, (P ∧ cint(X, Y, P))))	X intends to communicate P to Y

X and Y are agents, E is an action, P is a preposition, G is a goal.

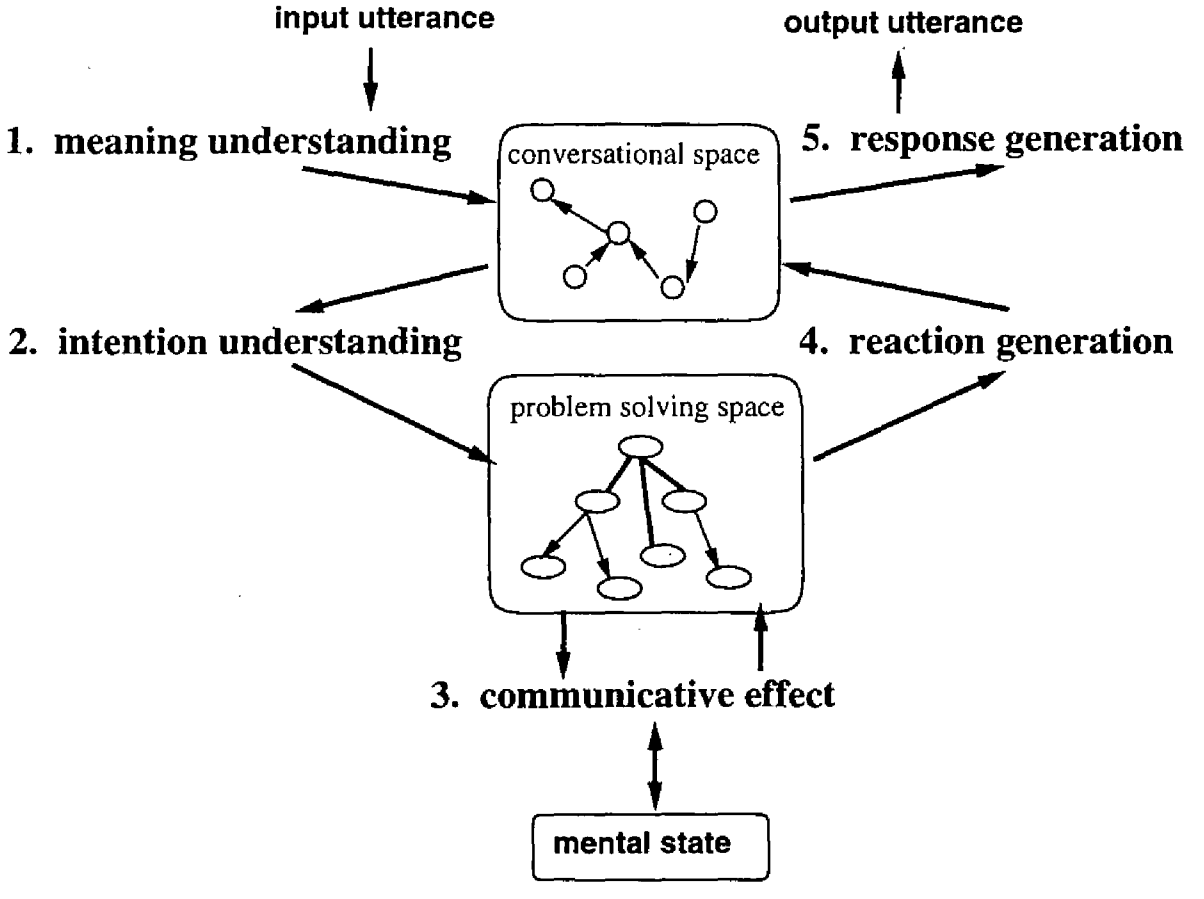


Figure 3.12: Five step modeling of dialogue understanding

steps.

After this, we explain each cognitive process.

3.3.1 Meaning understanding

The role of meaning understanding in spoken dialogue systems is to recognize the propositional content and illocutionary act of user's utterance. Concerning for extracting propositional content from utterance, it includes several problems in itself because recognition error is inevitable. Even so, we focus on the error in extracting propositional content as the one which robust parser cannot deal with by itself, because we concentrate on dialogue modeling in this chapter.

There are two kinds of errors which the robust parser cannot deal with:

(a) robust parser cannot generate semantic representation at all,

(b) no grammatical error was detected because of the replacement of the word with the word of the same syntactic and semantic category.

A possible treatment for these errors are:

(a) prompt re-enter the user by moving the processing at response generation step,

(b) put such error into the following processing because the error cannot detected in meaning understanding.

If there exists no error in robust parsing (or in case of (b)), we assume that there is no problem in extracting propositional content from utterance. For example, from an utterance "Register a meeting from 2 p.m.", the propositional content
$register([[start\_time, 2], [obj, meeting]])$
can be extracted and combined with literal illocutionary act:
do(U, lit-illoc(S, do(S, register([[start_time, 2], [obj, meeting]]), directive))).

The task of this step is to recognize illocutionary act from such semantic representation. The illocutionary act can be divided into two categories: *initiation* and *response*, in other words, forward looking functional statement and backward looking functional statement [2]. To recognize illocutionary act from the shared belief which consists of

[2]In natural dialogue, there also exist *followup* utterance which follows response, backchanneling, channel/external utterance which controls communication process itself. However, we concentrate on modeling the dialogue between spoken dialogue system and the user with some constraint. Therefore, the object here is limited in *initiation* and *response*.

1. Meaning understanding

   if shared_bel(S, U, do(U, express(S, int(U, do(S, E))))) = **true** ∨
   shared_bel(S, U, do(U, express(S, bel(U, P)))) = **true**
   then goto Intention understanding;
   else goto Response generation

2. Intention understanding

   if shared_bel(S, U, cint(U, S, int(U, do(U, S, G)))) = **true** ∨
   (shared_bel(S, U, do(U, S, G)) ∧
   (shared_bel(S, U, cint(U, S, int(U, do(S, E)))) ∨ shared_bel(S, U, cint(U, S, P))))
   = **true**
   then goto Communicative effect;
   else goto Response generation

3. Communicative effect

   if shared_bel(S, U, cint(U, S, int(U, do(U, S, G)))) = **true**
   then try(int(S, do(S, U, G)));
   if shared_bel(S, U, cint(U, S, int(U, do(S, E)))) = **true**
   then try(int(S, do(S, E)));
   if shared_bel(S, U, cint(U, S, P)) = **true**
   then try(bel(S, P));
   goto Reaction generation

4. Reaction generation

   if shared_bel(S, U, cint(U, S, int(U, do(U, S, G)))) = **true**
   then (cint(S, U, int(S, do(S, U, G))) ∧ cint(S, U, int(S, do(S, E)))) ∨
   (cint(S, U, ¬ int(S, do(S, U, G))) ∧ cint(S, U, bel(S, P)))
   if shared_bel(S, U, cint(U, S, int(U, do(S, E)))) = **true**
   then cint(S, U, done(S, E)) ∨ (cint(S, U, ¬ int(S, do(S, E))) ∧ cint(S, U, bel(S, P)))
   if shared_bel(S, U, cint(U, S, P)) = **true**
   then cint(S, U, int(S, do(S, U, do(S, E)))) ∨
   (cint(S, U, ¬ bel(S, P)) ∧ cint(S, U, bel(S, P')))
   goto Response generation

5. Response generation

   Ask back ∨ Generation by surface interaction rule ∨ Generation following the generated intention

try: predicate which tries to make given proposition true, express: expressing communicative intention, G(x,y) joint goal with y from x's viewpoint

Figure 3.13: Cognitive process in spoken dialogue

propositional content and literal illocutionary act of user's utterance is equal to determine whether the illocutionary act belongs to (1) initiation which user express to the system that he/she want the system to do something, or (2) response which user express his/her beliefs (see Figure 3.13).

1. User expresses that he/she intends system to do action E (int(U, do(S, E))).

2. User expresses that he/she believes proposition P (bel(U, P)).

Some literal illocutionary act can have the function both initiation and response. Furthermore, due to elliptical expression or substitutive expression of predicate part of utterance, in order to decide which function the utterance have, we need localized knowledge of dialogue. I our model, such problems are treated in conversational space (described in 3.4) by integrating the semantic representation into the previous local dialogue context.

After constructing shared belief by the process (1) or (2), the cognitive process goes forward next step: intention understanding.

3.3.2 Intention understanding

In order to succeed an illocutionary act in dialogue, speaker's intention and the intention that tries to communicate the speaker's intention must be communicated. Furthermore, such intention should be in the form of shared belief.

That the hearer understands what the speaker intends, in other words, success of speaker's illocutionary act means that "dialogue participants have shared beliefs of speaker's intention and the intention which tries to communicate the speaker's intention" Then, to what extent is the speaker's intention specified? In order for dialogue system to behave intelligently, the dialogue should be organized following task structure and each subdialogue should contribute to the subtask [21]. That is to say, the situation that speaker's plan is recognized by the hearer as the speaker's intention is preferable. Therefore, the purpose of this intention understanding step is to recognize speaker's intention of illocutionary act and speaker's plan.

However, there can be the situation that the hearer cannot narrow to one hypothesis of speaker's plan. The problem is how to maintain cooperative dialogue in such a situation. VanBeek and Cohen proposed the response generation mechanism in case of user's plan is ambiguous [37] (see 3.2.6). This method is useful when system must handle many plans.

We developed a method which maintain dialogue using surface interaction rule when user's plan cannot be identified [6]. In this method, plan recognition is made incrementally. That is to say, if the system can make shared belief of user's intention of trying to communicate the plan, the cognitive process goes to the communicative effect step. Otherwise, the process jumps to the response generation step. Plan recognition algorithm and how to generate response in the situation which speaker's plan cannot be identified is described in 3.5.

In Airenti's model [20], resolving indirect utterance is treated in this step by preparing some processing rule which explicitly make correspondence between indirect utterance to its illocutionary act. Contrary this, in our model, the variety of utterance are treated in conversational level plan in hierarchical plan definition.

3.3.3 Communicative effect

In this communicative effect step, mental state of the system is updated based on the result of previous intention understanding step. If new user's plan is recognized in intention understanding step, the system's processing are (1), and (2) or (3), corresponding to the input utterance. Otherwise, if user's plan is already recognized, the system's processing is (2) or (3). The detailed definition of achievable plan is explained in 3.5.

1. If user's intention is to propose an plan, and if the plan is achievable (there is no unachievable node in the decomposition set of the plan at problem solving space), then the system intends to do the plan (int(S, do(S, U, G))).

2. If user's intention is to make system to do an action, and if the action the proper move in intended plan (there is a link between intended plan and the action at problem solving space), and also it is achievable, then the system intends to do the action (int(S, do(S, E))).

3. If user's intention is to express his/her beliefs, and if the system does not have the beliefs which contradict the expressed beliefs, then the system believes the expressed beliefs (bel(S, P)).

After these updating procedure, the cognitive process goes to next reaction generation step.

3.3.4 Reaction generation

In this reaction generation step, system's intention is made from the result of intention understanding and communicative effect.

(1) If the system has an intention to do the recognized plan (or the plan has already shared), the system generates the intention following user's initiation. In this case, the system does not have to express having plan explicitly. By continuing cooperative and goal-oriented dialogue, it can implicitly show that it has shared belief of the user's plan.

(1-a) If the user's initiation is request for information, the system has the communicative intention to answer the user's question (cint(S, U, done(S, E))).

(1-b) If the user's initiation is request for action, the system checks out whether requested action is achievable in problem solving space.

(1-b-1) If the action is achievable, system does the action. Also it searches the next action E' that contributes the achievement of the plan, and has a communicative intention to communicate to do E'
(cint(S, U, int(S, do(S, E')))).
How to derive E' is explained in 3.5.

(1-b-2) If the action is not achievable, or it is not in a decomposition set of shared plan, the system has a communicative intention to communicate not to do E and the reason (cint(S, U, $\neg$ int(S, do(S, E))) $\wedge$ cint(S, U, bel(S, P))).

(2) If the system decides not to accept user's plan, that is, the plan is recognized but not accepted, the system has a communicative intention to communicate not to accept the plan explicitly and the reason, because there must exist obstacle in problem solving space about the plan
(cint(S, U, $\neg$ int(S, do(S, U, G))) $\wedge$ cint(S, U, bel(S, P))).

(3) In case user's utterance is expressing user's beliefs:

(3-a) If the system does not have a belief which contradicts user's expressed belief, the acceptance of user's belief is expressed implicitly by having an intention to do next action E which contributes plan achievement
(cint(S, U, int(S, do(S, E)))).

(3-b) If the system has a belief which contradicts user's expressed belief, the system has a communicative intention to express contradicted system's belief (int(S, U, ¬ bel(S, P)) ∧ cint(S, U, bel(S, P'))).

After these reaction generation procedure, the cognitive process goes to next response generation step.

3.3.5 Response generation

The process of response generation varies following which process activates the response generation.

1. If the process is activated by the failure of meaning understanding process, it generates simple ask back sentence "I beg your pardon?"

2. If the process is activated by the failure of intention understanding process, it generates the sentence using surface interaction rule which construct typical dialogue segment. The detail is described in 3.4.

3. If the process is activated after the processing of reaction generation, it generates the sentence using utterance templates according to each type of communicative intentions.

3.4 Conversational space

In order to understand illocutionary act of utterance, deal with elliptical expression, generate proper response to other participant's utterance, the cognitive process model described before needs to have a management mechanism of the progress of conversation. Many dialogue model which were proposed previously used the dialogue stack for these (or some of these) purposes. For example, Grosz et al. used the stack which contains present salient object, attributes, and discourse segment purpose of dialogue for the sake of focus management of their dialogue model [21]. In their dialogue model, they presupposes that the dialogue structure forms embedded structure in using stack management. But a natural dialogue does not necessarily make formal embedded structure. In addition, if we presupposes the existence of embedded structure, the dialogue management system must recognize such structure. In our model, we do not deal with embedded structure

explicitly. In stead of that, we presuppose the existence of an interaction unit as the minimal unit of dialogue and it is managed in **conversational space**.

An interaction unit consists of *initiation*, which appears at the top of interaction unit, *response*, which may appear after initiation successively, and *follow-up*, which may appear after response or follow-up. In some cases, follow-up utterance may be omitted. In other cases, before response utterance or at the place of it, another interaction unit may be inserted. The role of conversational space is to maintain the pattern of interaction unit and develop dialogue by exchanging the information to the process model (mainly, in the intention understanding step).

Conversational space is a kind of dynamic network growing with the development of dialogue. In conversational space, there are three types of nodes: **phrase node**, **instance node**, and **slot-filling node**. The definition of the node is same as the explanation in 3.4. The relation of elements in conversational space is shown in figure 3.14.

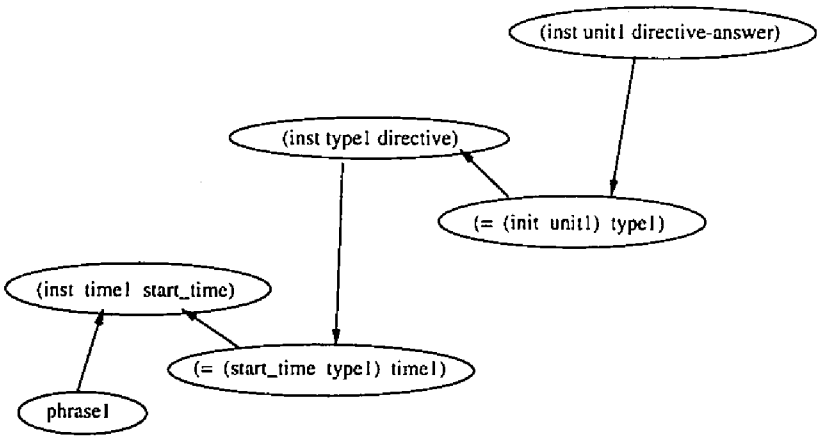


Figure 3.14: Relation of elements in conversational space

The processing algorithm of conversational space is as follows:

1. Introduce phrase node which corresponds the phrase in semantic representation of input utterance.

2. Introduce concept level instance node which shows the concept of phrase node, and make a link between them.

3. generate sentence level instance node which aggregates the concept nodes, generate slot-filling node which shows a relation between the sentence level instance node

and concept nodes (in grammatical term, it corresponds the case), and make links between them.

4. generate interaction level instance node which aggregates the sentence nodes, generate slot-filling node which shows a relation between the interaction level instance node and sentence nodes, and make links between them.

In this conversational space, not only objects, attributes, and discourse segment purpose, which Grosz et al. treated as the elements of focus, but also all the elements in interaction unit can use in processing of elliptical and referential expression according to the distance from present focus part of space. Also in this space, we can deal with surface interaction, e. g. clarification subdialogue, without consulting higher level knowledge. In addition, by combining with probabilistic understanding method described in , it can realize the identification of misrecognized word using forward dialogue context.

3.5 Problem solving space

We will feel spoken dialogue system as 'cooperative' if spoken dialogue system make proper answer and/or good suggestion. In order to generate such response, spoken dialogue system must recognize user's plan and select proper speech act as system's response.

Considering the plan-related function *move* and *precond*, which appear in the cognitive process model for spoken dialogue systems, and the task domain of our spoken dialogue system (group scheduling), we decided to use *Event hierarchy* [32] as a method of representing the plan. It is suitable for plan recognition as a process of gathering observed actions into an end plan. We call this network *Problem Solving Space* (PSS).

PSS is a static network that represents relationships between plan and subplans, and between plan and actions (Fig.3.15). This space is used in intention understanding step and reaction generation step.

Nodes of PSS represent plan (non-terminal nodes) and action (terminal nodes). Arcs mean abstraction relationship between plan and subplans (so called is-a relationship) and decomposition relationship between plan and actions. There are AND decomposition and OR decomposition in decomposition relationship.

We apply *minimal covering* method [32] for plan recognition in PSS. The basic point of this procedure is to find forest that covers all the subplans and actions previously achieved.

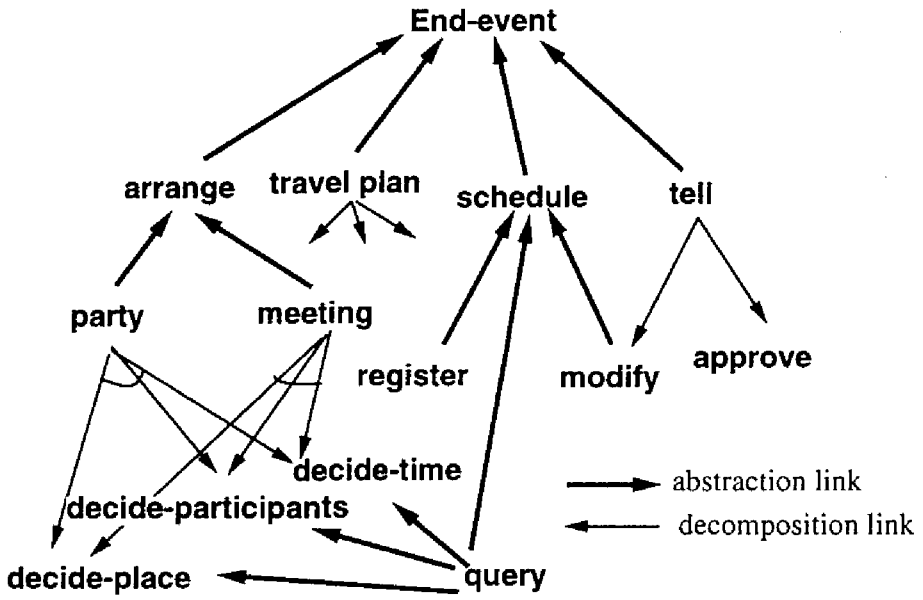


Figure 3.15: Problem Solving Space (part)

For example, after processing "Tell me an available time of dialogue group members tomorrow.", this utterance is located as an action of decide-time in PSS (Fig.3.15). This action is assumed to be a part of *arrange meeting* plan or a part of *arrange party* plan. And exceptionally, this action can be seen as *query* plan itself.

3.6 Example of system behavior

In this section, we show an example of system behavior. Figure 3.16 shows the example dialogue between user and personal schedule management system.

3.6.1 Processing first turn and plan recognition

As a result of robust parsing, we assume we can get following surface semantic representation.

shared_bel(S, U, do(U, lit-illoc(S, do(S,
register([[start_time, 2], [obj, meeting]])), directive)))

In meaning understanding step, the surface semantic representation is interpreted as a initiation of turn, because there is no element in conversational space and literal

U1: "Register a meeting from 2 P.M."
S2: "Until what time?"
U3: "Please modify until 5." (misrecognition: register → modify)
S4: "Do you want to modify it?"
U5: "Please register it."
S6: "Where is the place of the meeting?"
U7: "At small meeting room."(misrecognition: middle → small)
S8: "Small meeting room is not available at the time."
U9: "At middle meeting room."
S10: "So, a meeting from 2 P.M. to 4 P.M. at middle meeting room."
"Is that O.K.?"
U11: "Yes."

Figure 3.16: Example dialogue between user and personal schedule management system

illocutionary force of the surface semantic representation is directive. Then we can get following shared belief.

shared_bel(S, U, do(U, express(S,
int(U, do(S, register([[start_time, 2], [obj, meeting]])))))))

Next, in intention understanding step, as there is no shared plan between user and system, possible plan hypothesis, which the action do(S, register([[start_time, 2], [obj, meeting]])) is one of steps, is searched in problem solving space. The result of plan recognition is register_meeting_plan. Then we can get following two shared beliefs.

shared_bel(S, U, cint(U, S, int(U, do(U, S, register_meeting_plan)))) ∧
shared_bel(S, U, cint(U, S, int(U, do(U, S,
register([[start_time, 2], [obj, meeting]])))))

In communicative effect, the validity of the recognized plan is checked in problem solving space and current mental states. If there is no problem both, the system has following two intentions.

int(U, do(U, S, register_meeting_plan))) ∧
int(U, do(U, S, register([[start_time, 2], [obj, meeting]])))))

In reaction generation step, we use processing pattern of (1-b-1), described in previous section. Then we get another action of register_meeting_plan.

cint(S, U, int(S, do(U, inform_ref([end_time, S]))))

Finally, in response generation step, we use sentence template for inform_ref, in this case *motivateByInterrogative*, to make system's response.

3.6.2 Response generation in conversational space

We explain a response generation in conversational space using example here. Some kind of verb misrecognition cannot be detected at meaning understanding step. At intention understanding step, when an inconsistency of input illocutionary act with dialogue context is detected, a recovering sub-dialogue begins in conversational space.

From the viewpoint of interaction unit, dialogue context is initiation (u1) followed by initiation (s2). In the pattern of interaction unit, u3 must be the response to s2, or initiation which relates to S2. However, the recognition result of u3 does not suit both hypotheses.

shared_bel(S, U, do(U, express(S,
int(U, do(S, modify([[end_time, 4]])))))))

Then, supposing verb misrecognition, the system makes recovering sub dialogue pointing out the recognized verb:

S4: "Do you want to modify it?"

If user finds out system's misrecognition and says:

U5: "Please register it.",

then, the system replaces the verb (modify → register) at U3 in conversational space, deletes the interaction unit of recovering sub dialogue, and continues on dialogue.

On the other hand, if recovering sub dialogue fails to reach its purpose, current interaction unit (u1, s2, u3) is expired and system begins dialogue at the end of previous interaction unit.

In this case, u3 is interpreted as follows:

shared_bel(S, U, do(U, lit-illoc(S, bel(U, equal(end_time, 4)), assertive)))

After that, we can get s6 by this step.

meaning understanding: shared_bel(S, U, do(U, express(S, bel(U, equal(end_time, 4)))))
Intention understanding: shared_bel(S, U, cint(U, S, equal(end_time, 4)))
Reaction generation: cint(S, U, int(S, do(U, inform_ref([[place, P]]))))

In using Grosz et al's stack, the target word is limited. But in spoken dialogue system, any kind of words can be misrecognized and recovering method should be different following the type of misrecognition. Therefore, in spoken dialogue system, any element of previous context should be accessible. But the method of only recording previous conversation is not suitable for searching the target word. Then we express the dialogue context in a network manner which can be easily access the doubtful word. In addition, we limit the search space in interaction unit which is natural division of daily conversation in order to avoid keeping all the history of dialogue.

3.6.3 Intention understanding in problem solving space

The previous example shown in 3.6.2 is recovering method of misrecognition of verb. Here we show another error recovery example of content word misrecognition. If content word is replaced by the same categorical word, the misrecognition cannot detected until using dialogue context. [3]

Here, we explain a error detection and recovery in the processing of Problem Solving Space using example. After the example dialogue shown in 3.6.2, we assume following interaction occurs:

S6: "Where is it?"
U7: "At small meeting room."(misrecognition: middle → small)

At the end of the example shown in 3.6.2, as user's plan (register-meeting) was already presented in u1, we can suppose that plan recognition is in success. In such a case, the role of processing in Problem Solving Space is to assure that following interaction is for achieving subplans which contributes main user's plan, and the subplans can be achievable given instances. If system fails to achieve decide-place subplans because small meeting room is not available at the time, it makes reaction generation following the rule of (1-b-2) at 3.3.4:

[3]There remains another types of errors which cannot detected despite all these recovering method. Therefore, we have to add a confirmative interaction in spoken dialogue systems.

S8: "Small meeting room is not available at the time."

U9: "At middle meeting room."

If small meeting room is available, dialogue may continue on. Such type of misrecognition is to be recovered in confirmative interaction which mainly occurs at the end of the dialogue.

3.7 Discussion

In this section, we discuss about other approaches and about current problems of our approach.

In the previous works of written natural language understanding, many researches aim at the 'deep understanding', such as plan recognition in story understanding [39], and at the 'explanation' in text planning [40]. These studies do not treat the interaction in dialogue. Airenti et al.'s dialogue model [20] is aimed at the dialogue process from literal meaning to response generation form cognitive point of view. Basically, five steps division of our model follows Airenti's model. But because Airenti's model mainly take only one turn of dialogue, Airenti et al's model does not treat whole dialogue process as problem solving. And also Airenti's model cannot deal with ambiguity because it is basically rule based modeling.

Current problems of our approach are :

- As dialogue context is recorded mainly in mental state, there may be some difficulty in analyzing anaphoric expression only by the process in CS (for example, an anaphoric expression that indicates an entity that appeared before previous utterance). We need a simple heuristic method for analyzing anaphoric expression.

- User supposes to have the same (or subset) plan structure of system. If user have different way of solving problem, system cannot follow the dialogue. Then target dialogue is limited in this constraint.

- All the probabilities are settled by empirically at hand. In order to get valid probabilities by automatic learning, we need a moderate scale dialogue database that carefully tagged in detail. We are now trying to make such dialogue database.

3.8 Summary

In this chapter, we proposed a cognitive process model of spoken dialogue. We showed the necessity of two types of management processes: one is an understanding process which manages the subprocess from utterance understanding to response generation; the other is dialogue management process which aggregates the utterances into the discourse segment, and manages focus and intentions of dialogue. We combined these two management processes to stepwise cognitive process model which can deal with input errors caused by speech recognition errors. As a result, we integrated these three aspects which are treated independently in previous researches of spoken dialogue.

Chapter 4

Automatic Evaluation Environment for Spoken Dialogue Systems

4.1 Introduction

As many Spoken Dialogue Systems (SDSs) are implemented and demonstrated in several research organizations, the need for a total and efficient evaluation method of SDSs is more critical today than ever. However, as for interactive systems, previous evaluation methods are no longer adequate for evaluating some important points.

For example, the typical subsystem evaluation method that divides SDS into some subsystems in order to evaluate each subsystem independently, cannot measure total robustness of SDS. Also, the Input-Output pair evaluation method, that compares the input (typically speech) with the output (the result of dialogue processing, e.g. database records that satisfy the conditions involved in input), cannot measure interactive ability of SDSs because it uses prerecorded data.

In this chapter, we propose a new evaluation method for SDSs, that is system-to-system automatic dialogue with linguistic noise. This automatic evaluation method is the optimal method for measuring systems' ability of problem solving and their robustness. The proposed method is total, effective and an efficient way for improving the performance of SDSs by tuning parameters easily.

This chapter is organized as follows. We give an overview of the evaluation method of SDSs in 4.2. We propose a new evaluation environment in 4.3. Then, we show an example of evaluation in 4.4. Finally, we discuss our conclusions and describe the directions of future work in 4.5.

4.2 Survey of evaluation method for spoken dialogue systems

In this section, we present an overview of previous work concerning the evaluation of speech understanding systems, natural language understanding systems and dialogue systems. By surveying these works, we extract the defects of previous evaluation methods that we have to take into consideration in evaluating SDSs.

4.2.1 Subsystem evaluation

Previously, most SDSs were evaluated by the subsystem evaluation method. In this method, SDS is divided into some subsystems in order to evaluate each subsystem independently (Figure 4.1).

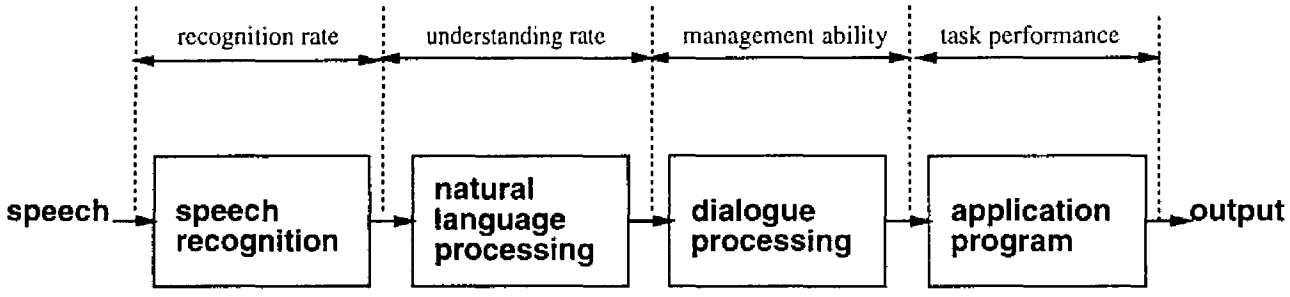


Figure 4.1: Concept of subsystem evaluation method

Concerning speech recognition subsystems, well established evaluation methods are word recognition rate or sentence recognition rate. Also in speech recognition subsystems, the difficulty of the target task domain is measured in terms of perplexity.

Concerning language processing subsystems, the developments of task independent evaluation methods are now in progress. One of these works is SemEval [41]. In SemEval, the meaning of a sentence is represented by predicate-argument structures, which provide semantically-based and application-independent measure.

However, the subsystem evaluation method cannot grasp the cooperation between subsystems. Some kinds of speech recognition error can be recovered by the linguistic knowledge. Also some kinds of syntactic / semantic ambiguity can be resolved by the contextual knowledge. The ability of dealing with such problems, that is *robustness*, is obtained by the cooperation of subsystems. But the subsystem evaluation method ignores

the possibility of these cooperations. Therefore, the subsystem evaluation method is inadequate. Total evaluation is essential for SDSs.

4.2.2 Input-output pair evaluation

In the ATIS (Air Traffic Information Service) domain, spoken dialogue systems are evaluated by language input / database answer pairs [42]. This allows us to evaluate total understanding in terms of getting the right answer for a specific task (Figure 4.2).

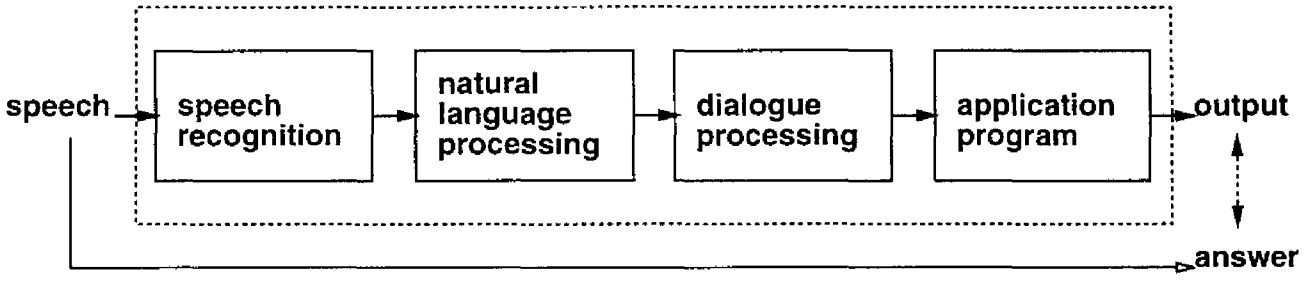


Figure 4.2: Concept of Input-Output pair evaluation

However, such evaluation methods cannot measure the interactive ability of SDSs, because they use prerecorded data. For evaluating interactive systems, prerecorded data cannot be used because the user's response determines what the system does next. The system's ability of interactive problem solving or of recovering from miscommunication cannot be evaluated by such methods. Therefore, we must enlarge the scope of evaluation still further to include interactive aspects of SDSs.

4.2.3 Evaluation by human judges

The alternative way of evaluating SDSs is using human judges (Figure 4.3). The evaluation is made by the task completion rate / time and by a questionnaire filled out by the human subjects.

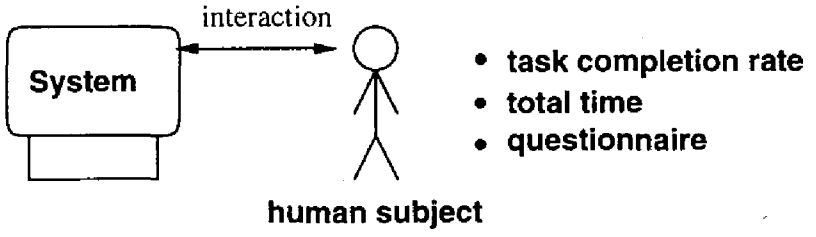


Figure 4.3: Concept of evaluation by human judges

This method is vital at the very last stage of evaluation of consumer products. But once evaluation includes human factors, it looses objectivity. Also human judgments take much time and are costly. At the early stage of research system development, we need more quick and low cost evaluation methods.

4.2.4 System-to-system automatic dialogue

A promising way of interactive system evaluation is through system-to-system automatic dialogue (Figure 4.4) ([43], [44], [45], [46]). The input and output of each system are both natural / artificial language texts. Dialogue is mediated by a coordinator program. The coordinator program opens communication channel at the beginning of dialogue, and closes it at the end of the dialogue. Also, the coordinator program records each utterance and judges whether the task has been successfully completed.

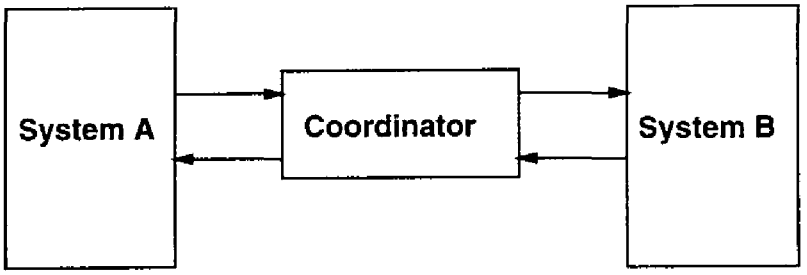


Figure 4.4: Concept of system-to-system automatic dialogue

In Japan, a system-to-system dialogue competition, *DiaLeague*, has taken place [46]. The task of this competition is shortest route search under incomplete and different route maps. The maps are similar to railway route maps. The map consists of stations and connections. But some connections between stations are inconsistent in the individual maps. Each system must find out the common shortest route from start station to goal station. Each system exchanges information about the map by using natural language text.

The purpose of this competition is to measure the system's ability of problem solving and the conciseness of the dialogue. In [46], they defined the ability of problem solving as the task completion rate. Also, they defined conciseness of dialogue as the number of content words. A smaller number of content words is preferable.

Such an automatic evaluation method can measure a total performance and an interactive aspect of dialogue system. Also, it is easy to test repeatedly. But this is for the

dialogue system using written text. We extend this evaluation method for SDSs in the next section.

4.3 Total and interactive evaluation of spoken dialogue systems

In this section, we describe our evaluation method of SDSs. First, we explain the concept of our method. Second, we describe an evaluation environment for system-to-system automatic dialogue with linguistic noise. Next, in order to make this evaluation independent of task, we define the concept of flexibility of an utterance and the flexibility of a dialogue. Finally, we discuss system parameters concerning the dialogue strategy.

4.3.1 System-to-system dialogue with linguistic noise

We have extended the concept of system-to-system automatic dialogue for the evaluation of SDSs (Figure 4.5). Random linguistic noise is put into the communication channel by the dialogue coordinator program. This noise is designed for simulating speech recognition errors. The point of this evaluation is to judge the subsystems' possibility to repair or manage such misrecognized sentences by a robust linguistic processor or by the dialogue management strategy.

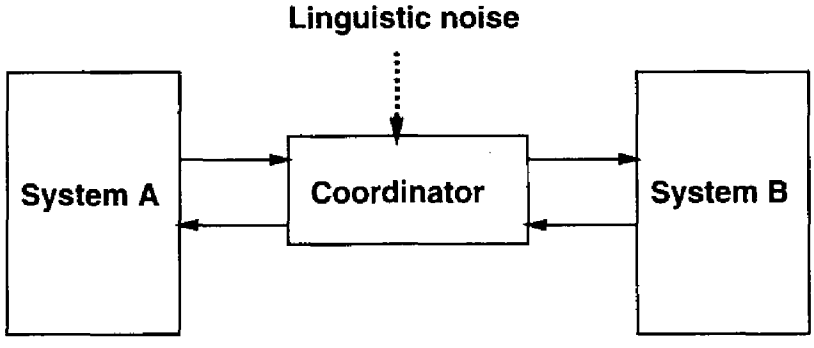


Figure 4.5: Concept of system-to-system dialogue with linguistic noise

With our method, the performance of a system is measured by the task achievement rate (ability of problem solving) and by the average number of turns needed for task completion (conciseness of dialogue) under a given recognition error rate.

4.3.2 Automatic dialogue environment

We have implemented an environment for the evaluation of automatic system-to-system dialogues. Figure 4.6 shows the concept of the environment.

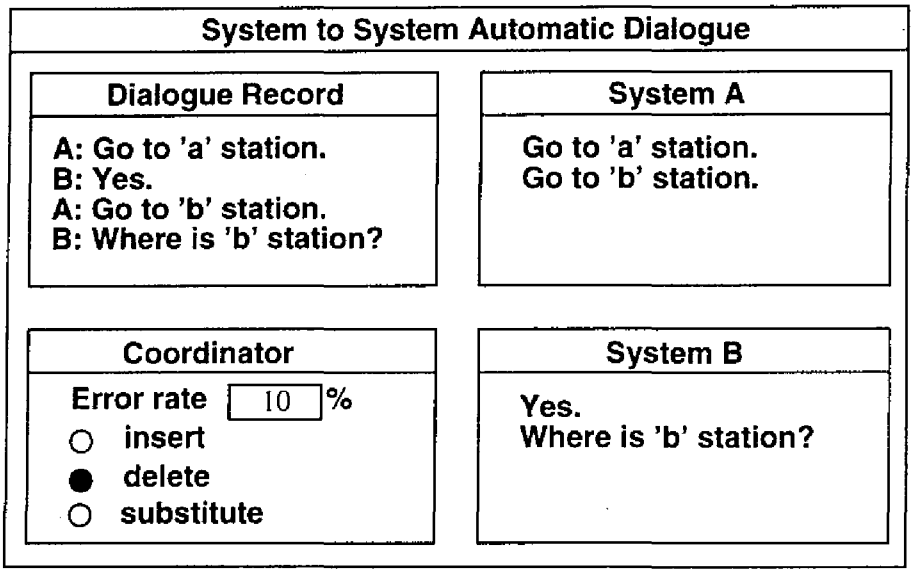


Figure 4.6: Concept of Evaluation environment

The environment consists of one coordinator agent and two dialogue agents. At the start of the dialogue, the coordinator sends a start signal to one of the dialogue agent. The dialogue agent who receives the start signal (system A) opens the dialogue. System A generates natural language text which is sent to the coordinator. The coordinator receives the text, puts linguistic noise into it at given rate, and passes the result to another dialogue agent (system B). System B has the next turn. The dialogue ends when one of the dialogue agents cuts the connection or when the number of turns exceeds the given upper bound.

The result of the dialogue is examined using logged data. In case both agents reach the same and correct answer, we regard the task problem as solved. The task achievement rate is calculated from the number of dialogues that reach the same and correct answer divided by the total number of dialogues. In addition, we assume that the conciseness of a dialogue can be measured by the average number of turns. This is because SDS puts a strain on the user each time he has to produce an utterance. Therefore we think smaller number of turns is preferable.

To make these values independent of task, we defined the flexibility of an utterance and the flexibility of a dialogue which we will describe in the following subsection.

4.3.3 Flexibility of utterance and dialogue

In order to make our evaluation method independent of task, we think another viewpoint must be added. In SDSs, language processing subsystems must deal with illegal input, such as ungrammatical sentences, sentences with unknown words, speech recognition errors, etc. However, if the correspondence between the sentence and its semantic representation is simple, then it is easy to recover errors or to collect partial results. In this situation, it needs less extra turns for the error recovery. As a result, the average number of turns is largely affected by the complexity of the correspondence between the sentence and its semantic representation.

The same is true concerning about dialogue management subsystems. A simple dialogue structure can reduce the extra turns for the error recovery.

In order to measure the complexities of these elements, we define for each task a *distance* from input utterance to its target semantic representation. We call this the *flexibility of an utterance*. The flexibility of an utterance is based on the distance from the result of the speech recognizer to the corresponding predicate-argument structure. The predicate-argument structure is a kind of frame representation of the sentence meaning. The main verb of the sentence always determines the frame name, that is, the predicate. Nouns or noun phrases are used to fill the value slot of arguments.

For defining the *flexibility of an utterance*, we have to specify the input of the language processing subsystems. Because the system deals with spontaneous speech, we can assume word lattice as input.

From the viewpoint of structural complexity of predicate-argument structure, we define the rank of flexibility of an utterance as follows:

1. **Ordered word sequence**

   An ordered word sequence is characterized by its content words and their order. An ordered word sequence corresponds to exactly one semantic representation (Figure 4.7). Typically, one of the content words corresponds to the slot of the verb in the predicate-argument structure, the rest of the content words simply fill the value slots of the arguments. Every word in a sequence has, in the given task, only one meaning entry in the dictionary. By this constraint, even if the speech recognizer outputs only content words and their order (e.g. by using word spotter), language processor can decide the proper slot of the words.

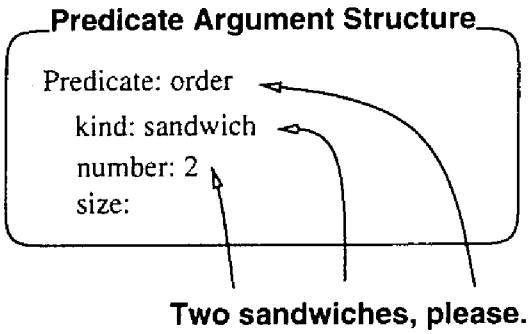


Figure 4.7: An example of ordered word sequence

2. **Simple sentence**

A simple sentence is defined as a type of sentences the semantic representation of which has only one predicate (Figure 4.8). Obviously, the ordered word sequence rank is a subset of the simple sentence rank. In a simple sentence, some content words can occupy a couple of value slot of the predicate-argument structure. Therefore, in contrast with ordered word sequence, it needs a structural information of the utterance in assigning the value of an argument. A possible parsing method used on this rank is keyword-driven parsing.

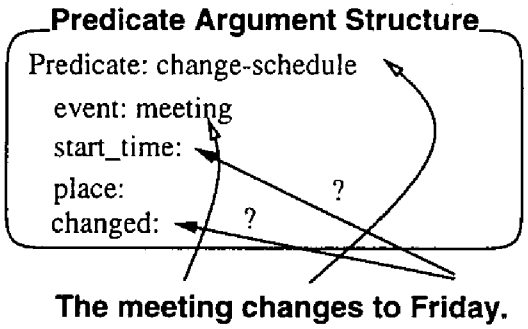


Figure 4.8: An example of simple sentence

3. **Complex sentence**

A Complex sentence is defined as a type of sentence whose argument value can be also a predicate argument structure (Figure 4.9). This definition is almost the same as in the linguistic terminology. It needs more structural information of the utterance in assigning the value of an argument than the simple sentence, because the possible value slots are increased in the predicate-argument structure.

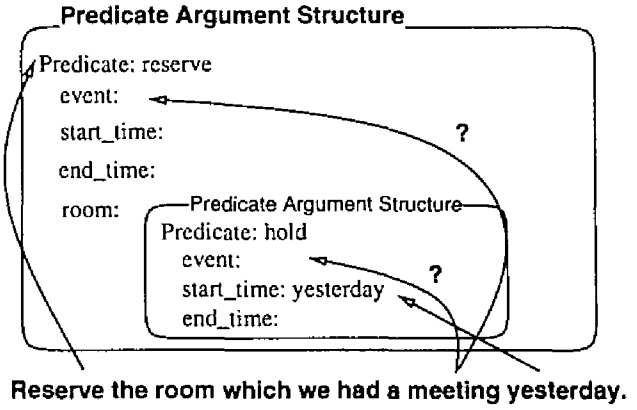


Figure 4.9: An example of complex sentence

However, if there exist tight dialogue level constraints, the *distance* seems to be diminished. We also define the *flexibility of a dialogue*. Considering the influence on language processing, we define ranks by using the following notion of dialogue model:

1. **Automaton dialogue model**

   Automaton dialogue models are in the class of regular grammars. The model strictly limits the flexibility of a dialogue by state transitions. Because of this limitation, expectations corresponding the next utterance can be used powerfully.

2. **Plan recognition based dialogue model**

   Plan recognition based dialogue models are in the class of context free grammars. For example, this model is implemented by event hierarchy ([32], [47]). The advantage of this rank of dialogue model is the tractability of focus shift which is limited in automaton dialogue models.

3. **Dialogue model in different knowledge**

   Dialogue models in different knowledge are proposed by Pollack [35] and also Grosz et al. [36]. Different knowledge means that participants of the dialogue do not share the same knowledge about plans and actions in the task domain. These models require the modeling of the user's mental state. Some studies employ first order predicate logic for representing the mental state. In general, the computational cost of this rank of model is higher than the other dialogue models. Also, the framework of this rank of dialogue model is mainly in tracing the change of the mental state. Such a framework is not suitable for the prediction of the next utterance.

4.3.4 Parameters of a dialogue strategy

We think that the dialogue strategy is another important factor for evaluating spoken dialogue systems. What types of feedback or what error recovery techniques are suitable using a given recognition error rate? What level of initiative is suitable for a given situation? These factors should be examined by overall system evaluation.

In our method, a dialogue strategy is represented by parameters of dialogue systems (e.g. level of initiative, type of feedback, frequency of confirmation, etc.). By changing these parameters, most suitable settings of parameters can be discovered in this evaluation environment.

4.4 Examples of evaluation by automatic dialogue

In this section, we show three examples of an overall system evaluation. First, we show the validity of the concept of evaluation using system-to-system dialogue by examining the effect of plan recognition in scheduling task domain. Second, we show the validity of evaluation of robustness by noisy dialogue simulation through examining the effect of confirmation utterance. In third example, we show the example of total evaluation of interactive system under automatic system-to-system evaluation with linguistic noise.

4.4.1 Examining the validity of evaluation by dialogue simulation

Purpose

In this example, we try to show the validity of the concept of evaluation using system-to-system dialogue by examining the effect of plan recognition in personal schedule management dialogue. Through this example, we try to draw the valid result to the following questions:

1. How does the number of turns in a dialogue depend on the use of plan recognition?
2. What type of answer can make dialogue concise?

Conditions

As a task domain of this experiment, we selected personal schedule management. We implemented heterogeneous dialogue agents: one is for the role of scheduling system (the

type of utterance of this *system agent* is shown in Table 4.2), and the other is for the role of user, which is designed by the analysis of dialogue corpus, who has a plan of solving scheduling problems (the type of utterance of this *user agent* is shown in Table 4.1.

Table 4.1: Utterance type of *system agent*

type	utterance
assert	(when)(where) de (event) wo touroku shitekudasai
modify	(event)wo(where)de(when)ni henkou shitekudasai
delete	(event)wo tyuushi shitekudasai
absent	(who)ha(event)wo kesski shimasu
reply	(when)desu (where)desu (who)desu hai
query of sch.	(who)wa(when)ni aiteimasuka (who)wa(when)ni yoteiha arimasuka
query of room	(when)(where)ha aiteimasuka
query of person	(who)no(when)no aiteiru jikanwo oshietekudasai (who)no(when)no yotei wo oshietekudasai
query of sch. of room	(when)ni aiteiru ROOM wo oshietekudasai (when)no(where)no yotei wo oshietekudasai (when)(where)no aiteiru toki wo oshietekudasai
declare	(event)wo settei shitai

The main purpose of this experiment is showing the effectiveness of system agents's plan recognition. Therefore, the number of turns are compared the plan recognition case with not-used case.

In order to make sure the result of comparing, we arranged two settings in *user agent's* utterance ((A) including unsettled attribute or (B) complete utterance), and two settings in *user agent's* planning ((1) including mistaken step for achieving the plan or (2) no-mistaken). Also, we varied the amount of schedule data.

In each case, we evaluated the effectiveness of plan recognition in dialogue system with a view of the number of turns. The dialogue situations are three scheduling tasks (schedule setting, modifying schedule and deleting schedule). The sample dialogue of schedule setting is shown in Fig sample:modify.

For the purpose of this example, showing the validity of evaluation using dialogue simulation, there exist no noise in communication channel.

Table 4.2: Utterance type of *user Agent*

type	utterance
reply	hai hai, aiteimasu iie, aiteimasen hai, arimasu iie, arimasen
query of sch.	nani wo touroku shimasuka itu desuka dokode desuka dareno desuka
confirm	(when)(where) de (event) wo touroku shimasu. iidesuka (event) wo (where) de (when) ni henkou shimasu. iidesuka (event) wo tyuushi ni shimasu, iidesuka (who) ga (event) wo kesseki ni shimasu. iidesuka
reply of sch.	(who) ha (when) (event) no yotei ga haitte imasu (who) ha (when) aite imasu
reply of room	(where) ha (when) (event) no yotei ga haitte imasu (where) ha (when) aite imasu

user(1):	I want to register a meeting.
system(2):	Yes.
user(3):	Tell me the available time of Prof. at 24th.
system(4):	Prof. can make time from 9 to 10, and from 11 to 17 at 24th.
user(5) :	Is Assistant Prof. free from 11 to 17 at 24th.
system(6):	No.
user(7):	Tell me the available time of All the members at 25th.
system(8):	All the members can make time from 10 to 12, and from 13 to 17 at 25th.
user(9):	Register a meeting from 10 to 12 at 25th.
system(10):	Where is it? Small meeting room is available.
user(11):	Register a meeting from 10 to 12 at 25th at small meeting room.
system(12):	A meeting from 10 to 12 at 25th at small meeting room. Is it O.K.?
user(13):	O.K.

Figure 4.10: An example dialogue of modifying schedule

Results

The results are shown below.

Table 4.3: The result of schedule setting case

system agent		user agent			
schedule data	plan recognition	A-1	A-2	B-1	B-2
little	not used	12.9	11.8	10.8	8.9
	used	11.4	11.1	9.2	8.9
	(improvement)	(11.2%)	(6.43%)	(14.8%)	(0%)
many	not used	19.5	17.1	15.7	13.4
	used	15.0	15.3	13.6	13.4
	(improvement)	(23.2%)	(10.4%)	(13.0%)	(0%)

Table 4.4: The result of modifying schedule case

system agent		user agent			
schedule data	plan recognition	A-1	A-2	B-1	B-2
little	not used	13.5	11.2	10.0	8.0
	used	10.6	10.3	9.0	8.0
	(improvement)	(21.5%)	(8.21%)	(10.0%)	(0%)
many	not used	18.1	16.7	14.4	12.6
	used	13.9	14.0	12.4	12.6
	(improvement)	(23.5%)	(16.0%)	(14.1%)	(0%)

Table 4.5: The result of deleting schedule case

system agent		user agent			
schedule data	plan recognition	A-1	A-2	B-1	B-2
little	not used	17.6	15.6	14.7	12.3
	used	12.9	14.2	12.3	12.3
	(improvement)	(26.6%)	(8.73%)	(16.4%)	(0%)
many	not used	8.1	8.1	6.8	6.8
	used	7.7	7.7	6.8	6.8
	(improvement)	(4.93%)	(4.93%)	(0%)	(0%)

The task achievement rate is 100% in all settings. The number of turns is less in the case of using plan recognition than in the case of not using, except in the case of user agent does not generate utterance including unsettled attribute and does not make a mistake in selecting the step for achieving plan, because there is no contribution point if plan

recognition. Compared with the setting of not using plan recognition, the maximum rate of decreasing the number of turns is 26.6% in the case of using plan recognition. Also, the rate is higher in the case of system agent manages much schedule data than in the case of one manages a little schedule data.

4.4.2 Examining the validity of robustness evaluation by noisy dialogue simulation

Purpose

In this example, we try to show the validity of robustness evaluation using noisy dialogue simulation by acquiring the following information:

1. How does the frequency and type of confirmation affect the task achievement rate of dialogue?

2. How does the frequency and type of confirmation affect the conciseness of dialogue?

Conditions

The task domain of this experiment is schedule management which is the same as the previous experiment. The dialogue systems for this experiment are implemented following the same principle of previous ones, but they are not identical. It is because dialogue systems for this experiment have to deal with communication errors. Table 4.6 shows the specification of the propositions which is used both *user agent* and *system agent*.

In this experiment, we examine the effect of confirmation policy toward solving the misunderstandings. The confirmation policy for this dialogue experiment can be defined as "When to confirm" and "What to confirm". We examined strategies as follows:

Confirming Strategies

Every-Time Confirmation: The *system agent* confirms to its *user agent* when new information is derives.

No-Confirmation: No confirmation even if the *system agent* is not confident about the information given from the *user agent*.

Confirm-When-Update: Confirm when the *system agent* is going to update its schedule knowledge.

Table 4.6: Specification of the propositions for the Group Scheduling Task

predicate	elements
assign	$from, fromtime, \{totime\|period\}, users, event$
delete	$from, fromtime, \{totime\|period\}, users, event$
change	$from, from, \{totime\|period\}, users, event, change - \$elements$
open	$from, fromtime, \{totime\|period\}, users$
plan	$from, to, period, event, users$
query_schedule	$from, to, users$
query_open	$from, to, users$

from : starting date

to : ending date

fromtime : starting time

totime : ending time

period : period in hours

event : event name for a schedule

users : participants for a schedule

$\$elements$: list of elements

notice: The brace "{}" means the alternative selection

Informing Strategies

Inform-All: The *system agent* informs to the *user agent* all the elements of the sentence.

Inform-Required: The *system agent* informs only required elements, that means it neglects the elements which has previously uttered.

The example of noisy dialogue simulation is shown in Figure 4.11.

user(1) :	I want to make a plan for a meeting about two hours, until the 24th.
user(1') :	*I want to make a plan for a meeting about three hours, until ....*
system(2):	A meeting, is it right?
user(3) :	Yes.
system(4):	From what day do you want to make a plan?
user(5) :	I want to make a plan from the 20th.
user(5') :	*I want to make a plan from the 10th.*
system(6):	From the 10th, is it right?
user(7) :	It is from the 20th.
user(7) :	It is 20th.
system(8):	From the 20th, is it right?
user(9) :	Yes.
system(10):	Who will take part in?
user(11):	It is the memer of Lab two.
system(12):	The member of lab two, is it right?
user(13):	Yes
system(14):	How about the schedule from ten o'clock on the 20th, about two hours?
user(15):	It is not open from 10 o'clock.
user(15):	*It is not open from 11 o'clock.*
system(16):	From 11 o'clock, is it right?

Figure 4.11: An example dialogue of noisy dialogue simulation
notice1 : the utterance in italic indicate the recognized ones, with communication errors
notice2 : The dialogue is translated from Japanese

Also, these policies are examined some settings of task. These are simple task, mode movement task, and difference of knowledge.

Simple Task: The *user agent* asks the *system agent* to assign an already determined schedule.

Mode Movement: The *user agent* consults the *system agent* to determine and to assign a schedule. The *user agent* has a choice of the next actions: offers to determine a suitable schedule; asks spare times; decides a schedule by himself.

Difference of Knowledge: incorporates knowledge errors in the Mode Movement Task.

Every experiment varies the rate of communication errors, represented as word recognition error rate, ranging from 0% to 50% with 10% as a step. In order to run an experiment on a particular dialogue strategy for a particular task, 100 dialogues for each range of communication error rate are simulated.

In each case, we evaluated the confirmation policy by average number of turns and task achievement rate. Average of turns is counted based on the dialogues in which both the user and the system agreed on assigning a schedule. This will show the length of the dialogues and can be used to measure the efficiency.

Task achievement rate is calculated using the following formula:

Task Achievement

$$Achieve(Task, Strategy, CommErr) = \frac{AchievedND(Task, Strategy, CommErr)}{ND(Task, Strategy, CommErr)}$$

$AchievedND(Task, Strategy, CommErr)$: Number of dialogues in which the same schedule was assigned in the $Task$ using $Strategy$ under communication error rate of $CommErr$.

$ND(Task, Strategy, CommErr)$: Number of dialogues in the $Task$ using $Strategy$ under communication error rate of $CommErr$.

In cooperative dialogues, the average number of turns should be lower and the task achievement should be higher.

Results

Simple task

The simple task is just to assign a specific schedule, already determined by the user, and there is no inconsistency between *user agent's* knowledge and *system agent's* knowledge. The *system agent'* recognizes what the *user agent* wants to do, and executes it. The plotted data for the average number of turns and the task achievement rate are summarized in figures 4.12 and 4.13.

Overall the rate of communication errors, the every-time—inform-all strategy shows the best results. As the simplicity of the experimental task, the loss of insufficiency was not raised compared to other strategies.

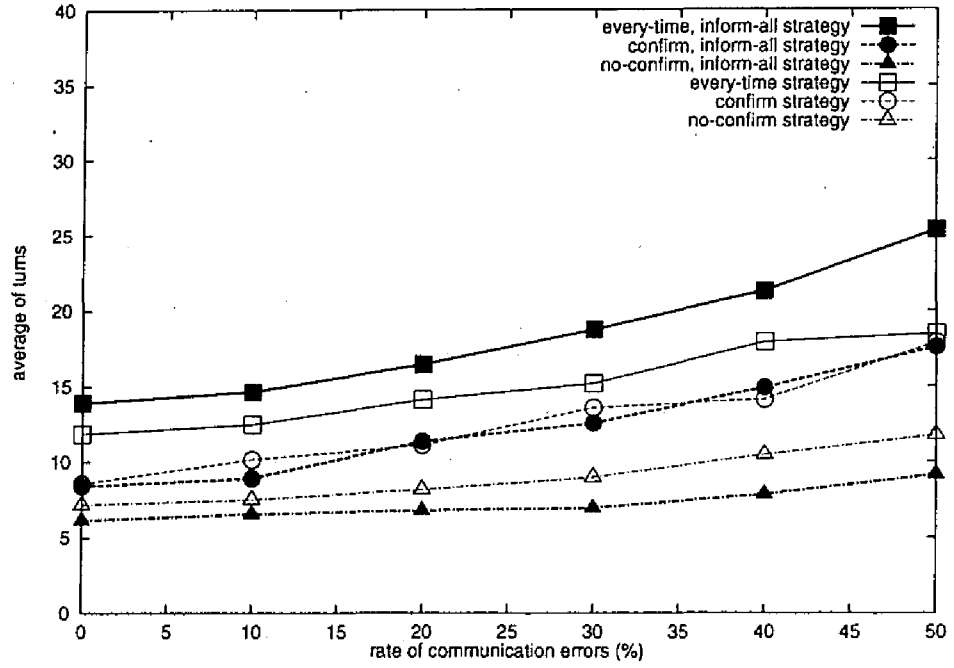


Figure 4.12: Simple Task : average number of turns

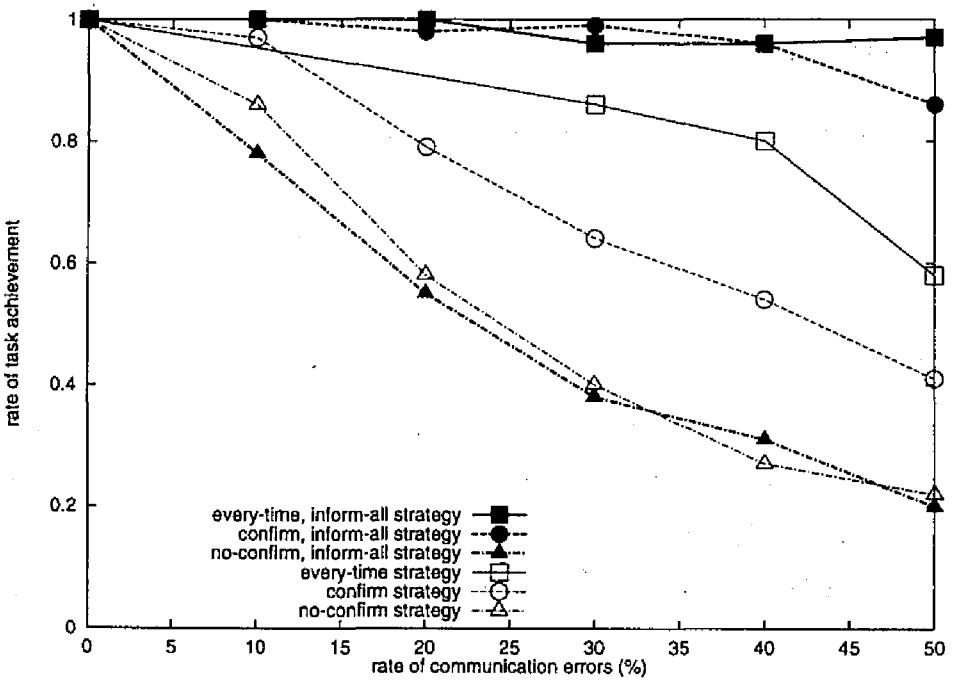


Figure 4.13: Simple Task : task achievement rate

On the other hand, the no-confirm strategies, both of the inform-all and inform-required strategies, were worst in all the range of communication errors. Among these two strategies, the inform-required strategy is somewhat better than the other.

The normal-confirm strategies are at the intermediate positions between above strategies, and the inform-all strategy shows better results than the inform-required strategy. Furthermore, the inform-all strategy shows almost the same as the every-time—inform-required strategy, which indicates that the confirming a lot with small information is nearly equivalent to the confirming sometimes, with providing all information.

Mode movement task

In the Mode Movement task, a user make a plan to assign a meeting, consulting the scheduling system. When planning a schedule, the user can select various actions, thus the system should follow what the user want to do next. The plotted data for the number of turns and the task achievement are summarized in figures 4.14 and 4.15.

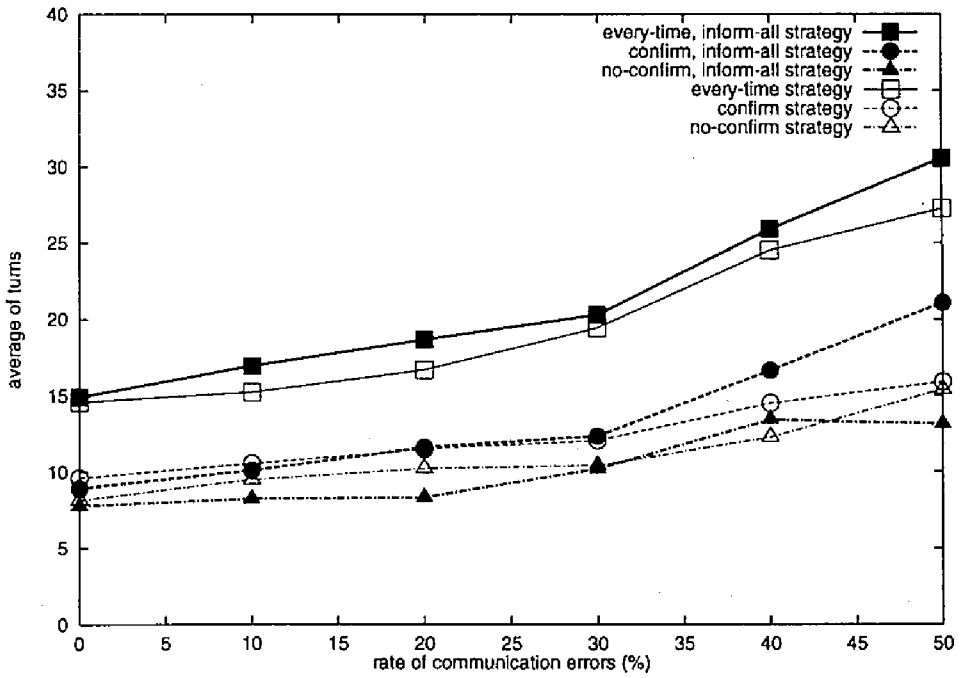


Figure 4.14: Mode Movement Task : average number of turns

In the error rate ranging from 0% to 20%, the every-time—inform-required strategy is superior to others, while the confirm—inform-all strategy is the best from 20% to 40%.

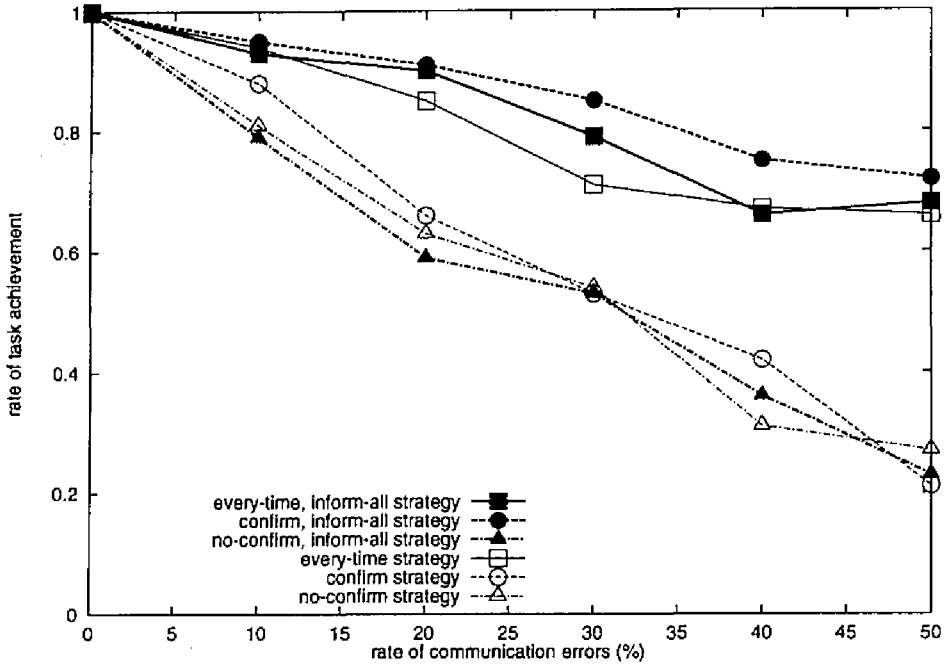


Figure 4.15: Mode Movement Task : task achievement rate

The every-time—inform-all strategy is almost intermediate among the strategies above.

The dialogue strategies are characterized in two classes:the class in which the task achievement is loosely lowered its performance and the number of turns is relatively raised, and the other class in which the task achievement rapidly decrease, and the number of turns is almost the same. the above strategies belong to the former, while the rest strategies belong to the latter.

The reason for the results is that the every-time strategies, at first seem to show better performance, can not neglect the communication errors: too often confirmation leads to misunderstandings and also results in the confusion of mode movement. In the normal-confirm—inform-all strategy, the proper confirmation, when the intention is fully recognized by the system, can modify its elements of information when pointed out by the user. Both of the no-confirm strategies and the normal-confirm—inform-required strategy is not good in task achievement (see figure 4.15), because the restricted elements make the confirmations meaningless. The system assumes that the mutual beliefs are already established in mentioned elements, which in turn miss the possibility to be corrected by the user.

Difference of Knowledge

The Difference of Knowledge Task is different from the Mode Movement Task in the incorporated difference of knowledge. The results are shown in figure 4.16 and 4.17.

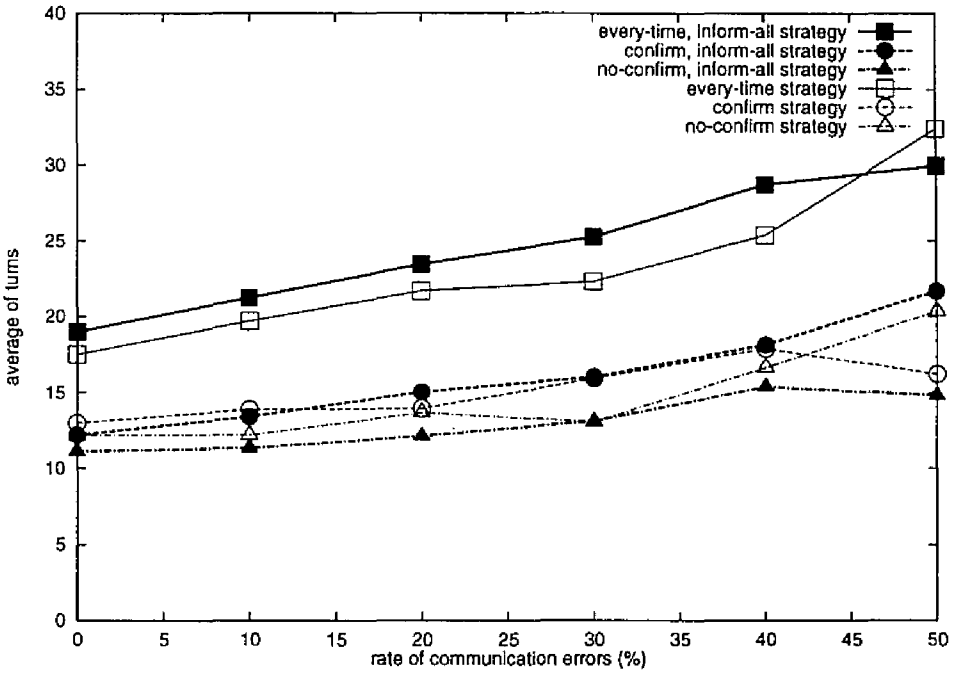


Figure 4.16: Difference of Knowledge Task : average number of turns

The normal-confirm—inform-all strategy shows the best results, while the every-time strategies are lower than this strategy. But the normal-confirm—inform-required strategy is as low as no-confirm strategy. The results are due to the fact, as argued above, that the communication errors would affect the ability to find out misunderstandings. Lots of confirmation result in the confusing of the system's mode, and lose the focus.

Other strategies are lower, as the results of Simple Task and Mode Movement.

Generalizations

- Generalizations about Tasks

  The Group Scheduling Task was selected as a simple planning task that incorporates the difference of knowledge and that requires negotiation to resolve it. The task can be described as resource allocation problems which is equivalent to many other planning tasks, though the representations are diverse. In the tasks for our

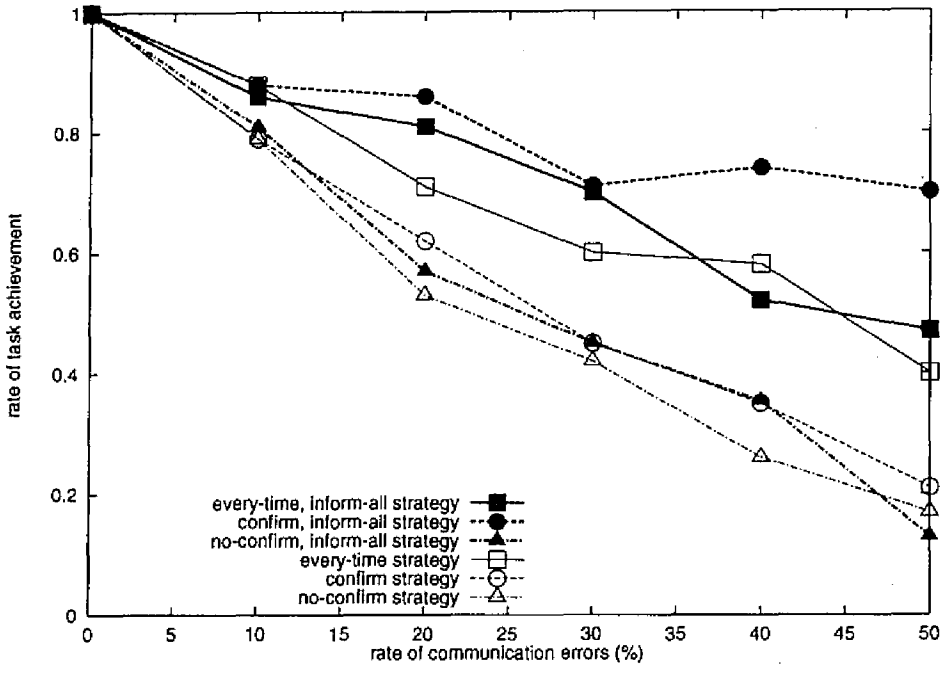


Figure 4.17: Difference of Knowledge Task : task achievement rate

experiments, we modified the degree of difference or conflicts in knowledge and the difficulties of its task achievement. These features can be easily applied to other tasks in other domains. For example in the domain of Route Search Problem, the knowledge can be represented as routes, thus the difference in knowledge can be described as those of map knowledge in dialogue agents.

- Generalizations about Dialogue System Model

  The dialogue system in the Group Scheduling Task is designed to evaluate the method to recover from misunderstandings through dialogue, and centered on how the system should lead the interaction with the user. The system was modeled with three modules as Evaluation, Problem Solving and Mediation, suitable to process and to manage various types of errors. This structure can be applied to other artificial agents or dialogue systems which are involved in negotiation to resolve conflicts either in the dialogue or knowledge level errors, or any other levels. Though the implemented planning mechanism is rather poor compared to other researches [43] [48] [36] [49] [50] [44] [45], the architecture, planning Finite States Automaton, is enough for our experiments, just to determine, and delete or assign a schedule. For other

agent models, the planning architecture might be required for some modifications.

The experimental results will extend to dialogues between humans and an actual speech dialogue system, as the dialogue model was based on the extensive analysis of dialogue corpus. In addition, simulated communication errors correspond to the errors occurred in a speech recognizer. Though the errors are limited to replacing in category or to missing and are restricted to the category of noun terms and post-positional particles, the results of the noised sentence is adequate for our experiment systems. The results would help develop a dialogue system to avoid or to resolve errors with the aid from its user by confirming. In applying to human–to–system dialogue, the system is required to process unknown words uttered by a user, which corresponds to parsing errors.

- Generalization about Dialogue Strategies

  Under human-to-system dialogue, the system can not predict what the user want to do for the first time, thus we assumed that a user would act rather randomly, while the system is equipped with fixed dialogue systems.

4.4.3 Examining the dialogue strategy

Purpose

On condition that dialogue system uses robust parse like described in chapter 2, the input utterance of spoken dialogue system occasionally lacks the part of utterance because of the recognition error or ellipsis. In such situations, if dialogue system asks back all the needed information for fulfilling the semantic representation of the utterance, the dialogue tends to be needlessly long and another recognition errors might be happened. But if system infers the lacked information from the dialogue context when system cannot grasp user's plan yet, the inference that compensate for the lost part might be wrong. If such a inference is made implicit, the dialogue continues with different belief, and it may be breakdown if the participants cannot find out when such an inconsistency yields.

In this example, we try to acquire the following information through examining automatic dialogue.

1. How does the number of turns in a dialogue depend on the number of speech recognition errors?

2. What type of dialogue strategy is most appropriate given a specific level of recognition accuracy?

Conditions

As a task domain of this experiment we selected shortest route search. This is identical with *DiaLeague* task. Each dialogue system has a route map of a railroad, which is somewhat different from the map the other system works with. Examples of a simplified map are shown in Figure 4.18. Some line might be cut, or some station name might be eliminated. The purpose of this task is to find out a shortest common route from start station to goal station.

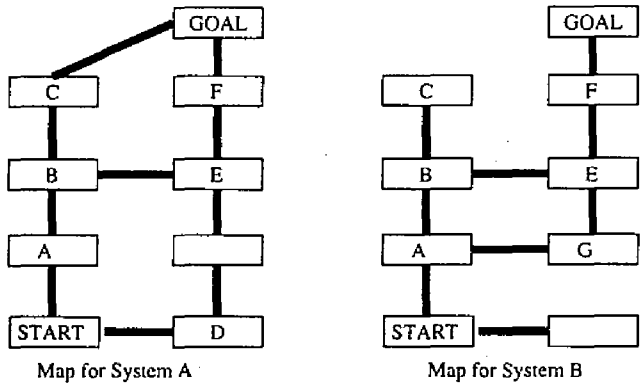


Figure 4.18: Examples of maps

In this experiment, we limited the flexibility of an utterance, ordered word sequence rank. The flexibility of dialogue was determined as automaton rank. Of course, a higher rank can be employed for this task. But examining the data of simulated human-human dialogue, we decide to model this task at the lowest rank of utterance and dialogue.

The employed automaton has 8 states and the number of sentence type is 9. Total words in this task are about 70 words. Among these words, 44 words are the name of stations in the map.

In this experiment, we set the error rate of speech recognition 10 % / 30 %. We simulate speech recognition errors by dropping one content word at given rate. This type of error reflects the errors often occurring in case of template matching in robust parsing.

We prepare two dialogue strategies: ask back type and infer type. The ask back strategy means that in case the system detects a speech recognition error, it asks again using the utterance "*Mou ichido itte kudasai.* (I beg your pardon?)" On the other hand,

the infer type strategy means that the system tries to infer what has to be said. The recovery strategy is that if a verb is missing, then the system infers the verb from the rest of the content words; if an important noun is left out, then the system gives up recovering and asks again. Multiplying these two conditions, we get four types of dialogue conditions. For each condition, we made three trials using the same map.

Results

In all 12 cases is reached the goal because of a simple rank of utterance and dialogue. Therefore, the task achievement rate is 100 %. Also, we counted the number of turns and calculated average turns to measure the conciseness of the dialogue. Table 4.7 shows the results of this experiment.

Table 4.7: Examining the dialogue strategy

error rate(%)	0	10	10	30	30
dialogue strategy	—	ask back	infer	ask back	infer
average number of turns	102	112	113	149	123

The first row shows that it takes 102 turns to solve the problem if there are no recognition errors. It is a baseline for other experiments. The second and third row show the results of examining the availabilities of two types of dialogue strategy respectively. They are achieved under 10 % recognition errors. It yields about 10 % of redundant interactions by recognition errors in both types. Even using different dialogue strategies, there is little difference in the average of turns. The rest shows the results under 30 % recognition errors. Another 33 % of redundant interactions yield from the ask back strategy, but 9 % from the infer strategy.

4.4.4 Examining the robustness of dialogue processing

Purpose

In this example, we try to acquire the following information through examining automatic dialogue.

1. How does the task achievement rate in a dialogue depend on the number of speech recognition errors?

2. To what extent does the proposed cognitive process model stand in speech recognition errors?

Conditions

As a task domain of this experiment we selected personal schedule management. This is identical with task described in 4.4.2.

In this experiment, because we want to concentrate on the problem on dialogue level, we omitted the problem of flexibility of an utterance. The flexibility of dialogue was determined as plan recognition rank described in chapter 3.

We set the error rate of speech recognition 10 % / 25 % / 40 %. Each experiment is done 16 or 17 times. We simulate speech recognition errors by replacing one content word at given rate. This type of error reflects the errors often occurring in case of template matching in robust parsing.

We define the breakdown of dialogue in two pattern considering the rational user's behavior to present computer systems. The first pattern of breakdown is a failure in confirmation. If there are more than two errors in confirmation utterance, we assume that user gives up the dialogue because user may select another way of communication instead of uncertain speech input. The second pattern of breakdown is an excess of number-of-turns limit. We defined the limit as twice as error free dialogue.

Results

Table 4.8 shows the results of this experiment.

Table 4.8: Examining the robustness of dialogue processing

error rate(%)	0	10	25	40
task achievement (%)	-	100	47	19
average turns (all)	7.0	7.7	9.7	10.0
average turns (success)	7.0	7.7	10.3	11.7

The task achievement rate rapidly fall down as the recognition error rate increases. On the contrary, average number of turns does not jump up because of the setting of this experiment.

4.5 Discussion

From first experiment, we can say that the harder the task is to achieve, the more efficient plan recognition is for dialogue system. The naturalness of the results shows the validity of dialogue system evaluation using dialogue simulation.

From second experiment, we can say that, overall the rate of communication errors, the every-time—inform-all strategy shows the best results. As the simplicity of the experimental task, the loss of insufficiency was not raised compared to other strategies. On the other hand, the no-confirm strategies, both of the inform-all and inform-required strategies, were worst in all the range of communication errors. Among these two strategies, the inform-required strategy is somewhat better than the other. The normal-confirm strategies are at the intermediate positions between above strategies, and the inform-all strategy shows better results than the inform-required strategy Furthermore, the inform-all strategy shows almost the same as the every-time inform-required strategy, which indicates that the confirming a lot with small information is nearly equivalent to the confirming sometimes, with providing all information.

From third experiment, using ordered word sequence rank and automaton rank, we can say that the infer type strategy is not effective at relatively low recognition rate. But if the recognition rate is high, the infer type strategy is more effective than the ask back strategy. It seems a natural conclusion. However, we think that this conclusion shows the validity of our evaluation method.

From fourth experiment, the robustness of cognitive process model proposed in chapter 3 is shown in relative low recognition error rate. The main reason of rapid decrease of task achievement rate is the rigid setting of the condition of dialogue breakdown. If user and system make redundant utterance, the misunderstanding may not pile up, then the misunderstanding can be easily resolved.

As a result, because we got reasonable result and analysis about experimental setting (or dialogue systems), then we can conclude that this evaluation environment is valid and suitable for the evaluation of robustness of interactive systems.

4.6 Summary

we proposed an evaluation environment for robust language / dialogue processing under interactive situation. We use this environment for evaluating proposed robust processing

method. In robust language processing, the parameter can be varied to make precision higher, that means restrain only plausible the output, or to make recall higher, that means generating the output anyway. On the other hand, in robust dialogue processing, the range of focused knowledge in inferring the lacked part of utterance can affect the task achievement rate or redundancy of dialogue. In order to determine such parameters, we need interactive dialogue situation. The recorded data cannot be used anymore for this purpose. Walker, Carletta, Hashida use system-to-system automatic dialogue as their method. We intend to evaluate our system's robustness to recognition errors or ill-formed sentences in spoken dialogue systems, we designed linguistic noisy channel in system-to-system automatic dialogue and establish evaluation methodology such interactive systems. In this environment, we examined the effectiveness of our robust processing methods.

Chapter 5

Conclusion

In this thesis, we described robust language processing and robust dialogue processing for spoken dialogue systems with general framework. Also, the generality and the effectiveness are examined under the interactive evaluation environment.

First, we explain the robust language processing method using path analysis of the semantic network, that can generate partial semantic representation toward the noisy input. Next, we propose a dialogue model that can choose the appropriate error recovery strategy following the result of understanding user's plan. Such robustness should be evaluated interactive environment under communication errors. We implemented system-to-system dialogue evaluation environment with linguistic noise, and showed the effectiveness of proposed robust language processing and dialogue model.

The followings are future works:

- Robust parser
  - develop a more flexible dialogue system that can predict the next user's utterance type and keywords.
  - apply more general semantic representation , such as WordNet.
  - getting into probabilistic measure in the plausibility of meaning representation.
- dialogue modeling
  - it is important to evaluate it by dialogue corpora.
  - unify the proposed model with mental state modeling
- evaluation method

- make a good linguistic generator for noise
- establish a measure of difficulties in each utterance and dialogue rank, which is similar to the perplexity in speech recognition.

Acknowledgments

The study has been accomplished in Kyoto University since I entered graduate school. I would like to express my sincere gratitude to Professor Shuji Doshita, my supervisor. Without his enlightening guidance and warmful support, I could not have completed the work.

I am grateful to Professor Katsuo Ikeda and Professor Michihiko Minoh for their invaluable comments to the thesis.

I also greatly appreciate Professor Toshiyuki Sakai, who introduced me to the research activities.

I would like to thank Professor Toyoaki Nishida of Nara Institute of Science and Technology (NAIST) for his fruitful comments and encouragement through the research.

I owe a great deal to the members of Professor Doshita's Laboratory. I had many suggestions from Dr. Tatsuya Kawahara and Dr. Atsushi Yamada (currently at ASTEM). I am also indebted to Mr. Taro Watanabe, Mr. Tetsushi Ikeda, Mr. Ikuo Azuma, and Mr. Takanobu Matsubara.

Finally, I wish to thank my family and friends for their support.

Bibliography

[1] T. Kawahara, S. Matsumoto, and S Doshita. A*-admissible context-free parsing on HMM trellis for speech understanding. In *Proc. of PRICAI92*, pages 1203–1208, 1992.

[2] T. Kawahara, T. Munetsugu, N. Kitaoka, and S. Doshita. Keyword and phrase spotting with heuristic language model. In *Proc. of ICSLP*, pages 815–818, 1994.

[3] R. M. Stern, W. H. Ward, A. G. Hauptmann, and J. Leon. Sentence parsing with weak grammatical constraints. In *Proc. of ICASSP87*, pages 380–383, 1987.

[4] W. H. Ward, A. G. Hauptmann, R. M. Stern, and T. Chanak. Parsing spoken phrases despite missing words. In *Proc. of ICASSP88*, pages 275–278, 1988.

[5] P. E. Fink and A. W. Biermann. The correction of ill-formed input using history-based expectation with applications to speech understanding. *Computational Linguistics*, 12(1):13–36, 1986.

[6] D. Stallard and R. Bobrow. The semantic linker – a new fragment combining method. In *Proc. of DARPA Human Language Technology Workshop*, pages 37–42, 1993.

[7] E. Jackson, D. Appelt, J. Bear, R. Moore, and A. Podlozny. A template matcher for robust NL interpretation. In *Proc. of DARPA Speech and Natural Language Workshop*, pages 190–194, 1991.

[8] T. Briscoe and J. Carroll. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. In S. Armstrong, editor, *Using Large Corpora*. The MIT Press, 1994.

[9] D. Hindle and M. Rooth. Structural ambiguity and lexical relations. In *Proc. of ACL*, pages 229–236, 1991.

[10] S. Miller, R. Bobrow, R. Ingria, and R. Schwartz. Hidden understanding models of natural language. In *Proc. of ACL*, pages 25–32, 1994.

[11] W. Ward. Understanding spontaneous speech: The PHOENIX SYSTEM. In *Proc. of ICASSP91*, pages 365–367, 1991.

[12] A. G. Hauptmann, S. R. Young, and W. H. Wayne. Using dialog-level knowledge sources to improve speech recognition. In *Proc. of AAAI*, pages 729–733, 1988.

[13] J. Pearl. *Probabilistic Reasoning in Expert Systems.* Morgan Kaufmann, 1988.

[14] L. M. R. Eizirik, V. C. Barbosa, and S. B. T. Mendes. A bayesian-network approach to lexical disambiguation. *Cognitive Science*, 17:257–283, 1993.

[15] E. Charniak and R. P. Goldman. A bayesian model of plan recognition. *Artificial Intelligence*, 64:53–79, 1993.

[16] M. Nagata and M. Suzuki. First steps towards annotating illocutionary force types to a bilingual dialogue corpus. In *Technical Report of JSAI (in Japanese)*, SIG-SLUD-9302-7, pages 49–56, 1993.

[17] C. J. Fillmore. The case for case. In E. Bach and R. T. Harms, editors, *Universals in linguistic theory.* Holt, Rinehart, & Winston, 1968.

[18] J. Peckham. Speech understanding and dialogue over the telephone: An ovrview of progress in the sundial project. In *Proc. of the 2nd European Conference on Speech Communication and Technology*, pages 1469–1472, 1991.

[19] A. Jönsson. A dialogue manager using initiative-response units and distributed control. In *Proc. of 5th Conference of the European Chapter of the Association for Computational Linguistics*, pages 233–238, 1991.

[20] G. Airenti, B. G. Bara, and M. Colombetti. Coversation and behavior games in the pragmatics of dialogue. *Cognitive Science*, 17:197–256, 1993.

[21] B. J. Grosz and C. L. Sidner. Attention, intention and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.

[22] J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proc. of 34th Meeting of the Assoc. for Computational Linguistics*, pages 62–70, 1996.

[23] S. R. Young. The minds systems: using context and dialogue to enhance speech recognition. In *Proc. of DARPA Speech and Natural Language Workshop*, pages 131–136, 1989.

[24] R. W. Smith and D. R. Hipp. *Spoken Natural Language Dialog Systems: A Practical Approach.* Oxford University Press, 1994.

[25] R. W. Smith, D. R. Hipp, and A. W. Biermann. An architecture for voice dialog systems based on prolog-style theorem proving. *Computational Linguistics*, 21(3):281–320, 1995.

[26] T. Kawahara and Y. Matsumoto. Robustness in spoken language processin. *Journal of IPSJ*, 36(11):1027–1032, 1995.

[27] J. Allen. *Natural Language Understanding.* The Benjamin/Cummings, 1995.

[28] M. E. Bratman, D. Israel, and M. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[29] J. L. Austin. *How to Do Things with Words.* Oxford University Press, 1962.

[30] J. R. Searle. *Speech Acts.* Cambridge University Press, 1969.

[31] R. C. Schank and C. K. (eds) Riesbeck. *Inside Computer understanding.* Yale University, 1981.

[32] H. A. Kautz. A circumscriptive theory of plan recognition. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication.* The MIT Press, 1990.

[33] A. W. Biermann, C. I. Guinn, R. Hipp, and R. W. Smith. Efficient collaborative discourse : A theory and its implementation. In *Proc. of DARPA Speech and Natural Language Workshop*, pages 177–181, 1993.

[34] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication.* The MIT Press, 1990.

[35] M. E. Pollack. Plans as complex mental attitudes. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication.* The MIT Press, 1990.

[36] B. J. Grosz and C. L. Sidner. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication.* The MIT Press, 1990.

[37] P. van Beek and R. Cohen. Resolving plan ambiguity for cooperative response generation. In *Proc. of IJCAI*, pages 938–944, 1993.

[38] M. Araki and S. Doshita. Cooperative spoken dialogue model using Bayesian network and event hierarchy. *Trans. of IEICE*, E78-d(6):629–635, 1995.

[39] E. Charniak. A neat theory of marker passing. In *Proc. of AAAI*, pages 584–588, 1986.

[40] I. Zukerman and R. McConachy. Being concise versus being shallow : Two competing discourse planning paradigms. In *Proc. of ECAI*, pages 515–519, 1994.

[41] R. C. Moore. Semantic evaluation for spoken-language systems. In *Proc. of ARPA Human Language Technology Workshop*, pages 126–131, 1994.

[42] L. Hirshman. Human language evaluation. In *Proc. of ARPA Human Language Technology Workshop*, pages 99–101, 1994.

[43] J. Carletta. *Risk-taking and Recovery in Task-Oriented Dialogue.* PhD thesis, University of Edinburgh, 1992.

[44] M. A. Walker. Dicourse and deliberation: Testing a collaborative strategy. In *Proc. of COLING94*, pages 1205–1211, 1994.

[45] M. A. Walker. Experimentallly evaluating communicative strategies: The effect of the task. In *Proc. of AAAI94*, pages 86–93, 1994.

[46] K. Hashida, et al. Dialeage. In *Proc. of the first annual meeting of the association for natural language processing (in Japanese)*, pages 309–312, 1995.

[47] M. Vilain. Getting serious about parsing plans: a grammatical analysis of plan recognition. In *Proc. of AAAI*, pages 190–197, 1990.

[48] J. Chu-Carroll and S. Carberry. A plan-based model for response generation in collaborative task-oriented dialogues. In *Proc. of AAAI*, pages 799–805, 1994.

[49] H. Iida and H. Arita. Natural language dialogue understanding on a four-typed plan recognition model. *Trans. of IPSJ (in Japanese)*, 31(6):810–821, 1990.

[50] J .F. Litman, D.J.and Allen. Discourse processing and commonsense plans. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*. The MIT Press, 1990.

List of Publications by the Author

Major Publications

[1] M. Araki, T. Kawahara, T. Nishida, and S. Doshita. Keyword-driven speech parser using dialog-level knowledge. In *Proc. of Pacific Rim Int'l Conf. on Artificial Intelligence*, pages 1025–1029, 1992.

[2] M. Araki, T. Kawahara, and S. Doshita. A keyword-driven parser for spontaneous speech understanding. In *Proc. of Int'l Sympo. on Spoken Dialogue*, pages 113–116, 1993.

[3] T. Kawahara, M. Araki, and S. Doshita. Reducing syntactic perplexity of user utterances with automaton dialogue model. In *Proc. of Int'l Sympo. on Spoken Dialogue*, pages 65–68, 1993.

[4] M. Araki, T. Watanabe, F. Quimbo, and S. Doshita. A cooperative man-machine dialogue model for problem solving. In *Proc. of Int'l Conf. on Spoken Language Processing*, pages 883–886, 1994.

[5] T. Kawahara, M. Araki, and S Doshita. Heuristic search integrating syntactic, semantic and dialog-level constraints. In *Proc. of IEEE Int'l Conf. Acoust., Speech & Signal Process*, volume 2, pages 25–28, 1994.

[6] M. Araki and S. Doshita. Cooperative spoken dialogue model using Bayesian network and event hierarchy. *Trans. of IEICE*, E78-d(6):629–635, 1995.

[7] M. Araki and S. Doshita. Automatic evaluation environment for spoken dialogue systems. In E. Mayer, M. Mast, and S. LuperFoy, editors, *Dialogue Processing in Spoken Language Systems*, pages 183–194. Springer, 1997.

[8] M. Araki and S. Doshita. Cognitive process modeling of dialogue for spoken dialogue systems (in Japanese). In *Journal of Natural Language Processing*, (submitted).

Technical Reports

[1] M. Araki, T. Kawahara, T. Nishida, and S. Doshita. A speech understanding system based on multiple key-word spotting and network-path analysis (in Japanese). In *IEICE Tech. Report*, SP91-94, pages 25–32, 1991.

[2] T. Munetsugu, T. Kawahara, M. Araki, and S. Doshita. Keyword spotting for spontaneous speech understanding (in Japanese). In *IEICE Tech. Report*, SP92-116, pages 7–14, 1993.

[3] N. Nukaga, M. Araki, T. Kawahara, and S. Doshita. Incremental sentence analysis with message passing on semantic network (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-9301-3, pages 19–26, 1993.

[4] N. Nukaga, M. Araki, T. Kawahara, and S. Doshita. Incremental sentence understanding using hierarchcally structured semantic network (in Japanese). In *IEICE Tech. Report*, SP93-126, pages 63–70, 1994.

[5] M. Araki and S. Doshita. A designing method of spoken dialogue system based on the constraints on dialogue and utterance (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-9502, 1995.

[6] M. Araki, I. Azuma, G. Tanaka, and S. Doshita. Spoken dialogue database and case-based semantic analysis (in Japanese). In *JSIP Tech. Report*, 95-SLP-8-5, 1995.

[7] T. Kawahara, M. Araki, and S. Doshita. Comparison of parsing and spotting approaches for spoken dialogue understanding. In *Proc. ESCA workshop on Spoken Dialogue Systems*, pages 21–24, 1995.

[8] M. Araki and S. Doshita. Automatic evaluation environment for spoken dialogue systems. In *Proc. of ECAI Workshop on Dialogue Processing in Spoken Language Systems*, pages 8–12, 1996.

[9] T. Ikeda, T. Sasaki, M. Araki, and S. Doshita. Multimodal infomation understanding by integrating speech, gesture and diagram (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-9603-3, 1997.

[10] T. Watanabe, M. Araki, and S. Doshita. Conflict resolution in difference of knowledge and recognition errors in dialogue system (in Japanese). In *JSIP Tech. Report*, 96-SLP-14-7, 1996.

[11] M. Araki, et al. Progress report of the discourse tagging working group. In *JSAI Tech. Report*, SIG-SLUD-9701-6, 1997.

[12] M. Araki, T. Watanabe, and S. Doshita. Evaluating dialogue strategies under various communication errors. In *Proc. of IJCAI Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems*, pages 13–18, 1997.

Oral Presentations

[1] M. Araki, T. Nishida, and S. Doshita. Understanding spoken japanese sentence using conceptual information (in Japanese). In *Proc. Annual Convention JSAI*, 13-4, pages 393–396, 1990.

[2] M. Araki, T. Saito, K. Satoh, T. Nishida, and S. Doshita. Speech understanding using dialogue structure and concept of words (in Japanese). In *Proc. Annual Convention JSIP*, 4C-1, pages 61–62, 1991.

[3] M. Araki, H. Ishibashi, T. Nishida, and S. Doshita. Analysis of spoken sentences integrating phonetic and syntactic information based on dempster-shafer's theory (in Japanese). In *Proc. Annual Convention JSAI*, 15-2, pages 591–594, 1991.

[4] M. Araki, T. Nishida, and S. Doshita. A speech understanding system based on multiple key-word spotting and network-path analysis (in Japanese). In *Proc. Annual Convention JSIP*, 6N-4, pages 151–152, 1992.

[5] M. Araki, T. Kawahara, and S. Doshita. Keyword-driven speech parser using multi-level knowledge source (in Japanese). In *Proc. Annual Convention JSAI*, 6b-1, pages 559–562, 1992.

[6] T. Kawahara and M. Araki. Comparison of left-to-right A* search and keyword-driven search (in Japanese). In *Symposium on Continuous Speech Recognition*, SPREC91-2, pages 29–32, 1992.

[7] N. Nukaga, M. Araki, T. Kawahara, T. Nishida, and S. Doshita. Utilization of semantic constraint to search of speech understanding (in Japanese). In *Proc. Annual Convention JSIP*, volume 3, pages 77–78, 1992.

[8] M. Araki, T. Munetsugu, T. Kawahara, and S. Doshita. Spontaneous speech understanding by semantic-driven parser (in Japanese). In *Proc. Annual Convention JSIP*, volume 2 of *1E-4*, pages 231–232, 1993.

[9] N. Nukaga, M. Araki, T. Kawahara, T. Nishida, and S. Doshita. Incremental sentence analysis with message passing on semantic network for spoken dialogue understanding (in Japanese). In *Proc. Annual Convention JSAI*, 1993.

[10] T. Kawahara, M. Araki, T. Munetsugu, and S Doshita. Comparison of left-to-right parser and keyword-driven parser for spontaneous speech understanding (in Japanese). In *Proc. Meeting Acoust. Soc. Japan*, 3-4-9, spring 1993.

[11] T. Kawahara and M. Araki. A new computational model for spontaneous speech understanding (in Japanese). In *Symposium on Spontaneous Speech Processing*, SPREC93-1, pages 49–53, 1993.

[12] T. Kawahara, M. Araki, K. Kohda, N. Nukaga, and S. Doshita. On the use of dialogue-level knowledge to improve speech recognition (in Japanese). In *Proc. IEICE Conf.*, SA-6-3, autumn 1993.

[13] T. Kawahara, M. Araki, and S Doshita. Syntactic prediction of next user utterances with dialogue state transition model (in Japanese). In *Proc. Meeting Acoust. Soc. Japan*, 1-18-12, autumn 1993.

[14] M. Araki and S Doshita. Presumption of meaning of utterance and unknown words using dialogue case base (in Japanese). In *Proc. Annual Convention JSIP*, volume 3 of *4G-1*, pages 155–156, 1994.

[15] M. Araki, T. Watanabe, and S. Doshita. Modeling of cooperative problem solving by dialogue (in Japanese). In *Proc. Annual Convention JSIP*, pages 587–590, 1994.

[16] F. Quimbo, M. Araki, and S. Doshita. Topic identification and prediction based on the domain plan and the discourse structure. In *Proc. Annual Convention JSAI*, pages 591–594, 1994.

[17] J. Shih, M. Araki, and S. Doshita. Proposal of a negotiation protocol for multi-user scheduling system. In *Proc. Annual Convention JSIP*, volume 3, pages 233–234, 1994.

[18] S. Nakagawa, M. Araki, and S. Doshita. Incremental semantic analysis from fragments of utterance (in Japanese). In *Proc. Annual Convention JSIP*, 7R-3, pages 223–224, 1994.

[19] J. Shih, M. Araki, and S. Doshita. Proposal of agent communication protocol for multi agent scheduling system (in Japanese). In *Proc. Annual Convention JSIP*, volume 4N-5, pages 289–290, 1994.

[20] S. Nakagawa, M. Araki, and S. Doshita. A method of semantic analysis from head of utterance based on extraction semantic rift (in Japanese). In *Proc. Annual Convention JSIP*, 4G-2, pages 157–158, 1994.

[21] T. Ikeda, M. Araki, and S. Doshita. Intention understanding using bayesian network (in Japanese). In *Proc. Annual Convention ANLP*, pages 69–72, 1995.

[22] I. Azuma, M. Araki, and S. Doshita. Collection of dialogue data and statistical analysis of semantic information (in Japanese). In *Proc. Annual Convention JSAI*, pages 541–544, 1995.

[23] T. Watanabe, M. Araki, and S. Doshita. Negotiation model in goal oriented dialogue (in Japanese). In *Proc. Annual Convention ANLP*, pages 349–352, 1996.

[24] J. Nomura, M. Araki, and S. Doshita. Implementation of incremental utterance understanding system using plan recognition of phrase. In *Proc. Annual Convention JSAI*, pages 415–418, 1996.

[25] T. Ikeda, M. Araki, and S. Doshita. Dictation of lecture style speech using diagrammatic information (in Japanese). In *Proc. Annual Convention JSIP*, 7N-6, pages 357–358, 1996.

[26] I. Azuma, M. Araki, and S. Doshita. Semantic tagging to dialogue database (in Japanese). In *Proc. Annual Convention JSIP*, 7L-6, pages 115–116, 1996.

[27] J. Nomura, M. Araki, and S. Doshita. Mulitimodal drawing system using speech, mouse, and keyboard. In *Proc. Annual Convention JSAI*, pages 388–391, 1997.

Appendix

I: List of 50 Sample Grammatical Sentences

Note: English translation is for reference.

1. 明日の２時から３時まで２研で音声研究会を開きたい。
   asu no niji kara saNji made nikeN de oNsee keNkyuukai o hiraki tai
   (I want to hold a speech group meeting at room No. 2 from 2 to 3 o'clock tomorrow.)

2. はい。
   hai
   (O.K.)

3. １０月３０日に人工知能学会の原稿の締切があります。
   juugatsu saNjuunichi ni jiNkoochinoogakai no geNkoo no shimekiri ga arimasu
   (Oct. 30th is the deadline of the paper for Society of Artificial Intelligence.)

4. 結構です。
   kekoo desu
   (No, thank you.)

5. ２０日から２３日まで情報処理学会で名古屋に出張します。
   hatsuka kara nijuusaNnichi made joohooshorigakai de nagoya ni shuchoo shimasu
   (I will make a business trip to Nagoya from the 20th to the 23rd for a conference of Information Processing Society.)

6. あさっての午前１０時から１２時までセミナー室で打合せを行う。
   asate no gozeN juuji kara juuniji made seminaashitsu de uchiawase o

okonau
(I will have a meeting at the seminar room from 10 to 12 A.M. on the day after tomorrow.)

7. いいえ、１２時までです。
iie juuniji made desu
(No, until 12 o'clock.)

8. そうです。
soodesu
(That's right.)

9. 明日の９時から会議を行う。
asu no kuji kara kaigi o okonau
(I will have a meeting at 9 o'clock tomorrow.)

10. １１時までです。
juuichiji made desu
(Until 11 o'clock.)

11. 大会議室です。
daikaigishitsu desu
(At the large meeting room.)

12. １７日に休暇をとりたい。
juunananichi ni kyuuka o tori tai
(I want to take a holiday on 17th.)

13. じゃあ、やめます。
jaa yame masu
(Then, I'll cancel the request.)

14. ２５日の１３時から１６時まで１研で予算会議を開きたい。
nijuugonichi no juusaNji kara juurokuji made ichikeN de yosaNkaigi o hiraki tai
(I want to have a budget meeting at room No. 1 from 13 to 16 o'clock on the 25th.)

15．１５時までなら大丈夫ですか？
juugoji made nara daijoobudesuka
(Can you attend if it is until 15 o'clock.)

16．来週の火曜日の１４時から１時間３講で定性推論研究会を行いたい。
raishuu no kayoobi no juuyoji kara ichiji kaN saNkoo de teeseesuiroN keNkyuukai o okonai tai
(I want to have a qualitative reasoning group meeting for an hour from 14 o'clock next Tuesday.)

17．１６時以降のスケジュールはどうなっていますか？
juurokuji ikoo no sukejuuru wa doonateimasuka
(What is the schedule after 16 o'clock ?)

18．では、１６時からにします。
dewa juurokuji kara ni shimasu
(Then, I will start it at 16 o'clock.)

19．１３日は阪大での自然言語処理研究会に出席する。
juusaNnichi wa haNdai deno shizeNgeNgoshori keNkyuukai ni shuseki suru
(On the 13th, I will attend the natural language group meeting at Osaka Univ.)

20．１０時以降ならあいていますか？
juuji ikoo nara aiteimasuka
(Are you free after 10 o'clock ?)

21．では、１０時から出かけます。
dewa juuji kara dekake masu
(Then, I will go out at 10 o'clock.)

22．あすの分科会を午後３時からに変更する。
asu no buNkakai o gogo saNji kara ni heNkoo suru
(Change the time of tomorrow's sub-committee meeting to 3 P.M.)

23．１８日の会議の場所を第２演習室に変更する。
juuhachinichi no kaigi no basho o dainieNshuushitsu ni heNkoo suru
(Change the place of the meeting on the 18th to seminar room No. 2.)

24. いいえ、第２演習室です。
iie dainieNshuushitsu desu
(No, at seminar room No. 2.)

25. 明日の音声研究会を中止する。
asu no oNsee keNkyuukai o chuushi suru
(Cancel tomorrow's speech group meeting.)

26. ああ、あさってでした。
aa asate deshita
(Oh, it is the day after tomorrow.)

27. １５日の特別講演は中止になりました。
juugonichi no tokubetsukooeN wa chuushi ni narimashita
(The special lecture on the 15th is canceled.)

28. 特別講演はいつですか？
tokubetsukooeN wa itsu desuka
(When will the special lecture be held ?)

29. それが中止になりました。
sore ga chuushi ni narimashita
(It is canceled.)

30. あさっての会議を１６時からに変更する。
asate no kaigi o juurokuji kara ni heNkoo suru
(Change the time of the meeting on the day after tomorrow to 16 o'clock.)

31. では、１５時からにします。
dewa juugoji kara ni shimasu
(Then, change it to 15 o'clock.)

32. 金曜日の予算会議の場所を小会議室に変更したい。
kiNyoobi no yosaNkaigi no basho o shookaigishitsu ni heNkoo shitai
(I want to change the place of the budget meeting next Friday to the small meeting room.)

33. では、そのままでいいです。
dewa sonomama de iidesu
(Then, leave it as it is.)

34. ２３日の音声研究会を９時からに変更したい。
nijuusaNnichi no oNsee keNkyuukai o kuji kara ni heNkoo shitai
(Change the time of the speech group meeting on the 23rd to 9 o'clock.)

35. １１時以降の予定は何がありますか？
juuichiji ikoo no yotee wa naniga arimasuka
(What are the plans after 11 o'clock ?)

36. では、１３時から１５時までに変更します。
dewa juusaNji kara juugoji made ni heNkoo shimasu
(Then, make it 13 to 15 o'clock.)

37. 今日のセミナーは何時からですか？
kyoo no seminaa wa naNji kara desuka
(What time does today's seminar start ?)

38. １５日の勉強会は何時からですか？
juugonichi no beNkyookai wa naNji kara desuka
(What time will the seminar on the 15th start ?)

39. 何時までですか？
naNji made desuka
(What time will it end ?)

40. パリ出張は何日から何日までですか？
pari shuchoo wa naNnichi kara naNnichi made desuka
(On which days is your business trip to Paris scheduled ?)

41. 自然言語処理研究会の原稿の締切はいつですか？
shizeNgeNgoshori keNkyuukai no geNkoo no shimekiri wa itsu desuka
(What day is the deadline of the paper for the natural language group ?)

42. あさっての会議はどこでありますか？
asate no kaigi wa dokode arimasuka
(Where will the meeting on the day after tomorrow be held ?)

43. 今日の予定は何がありますか？
kyoo no yotee wa naniga arimasuka
(What are today's plans ?)

44. １６日の午後の予定はどうなっていますか？
juurokunichi no gogo no yotee wa doonateimasuka
(What are the plans for the afternoon of the 16th ?)

45. あすの研究会はどこで？
asu no keNkyuukai wa dokode
(Where will the group meeting be held tomorrow ?)

46. １６日の発表会は何時から？
juurokunichi no hapyookai wa naNji kara
(What time will the presentation on the 16th start ?)

47. ああ、１７日でした。
aa juunananichi deshita
(Oh, it is on the 17th.)

48. ５日の定性推論研究会は何時から？
itsuka no teeseesuiroN keNkyuukai wa naNji kara
(What time will the qualitative reasoning group meeting on the 5th start ?)

49. じゃあ、勘違いでした。
jaa kaNchigai deshita
(Then, I was wrong.)

50. 今日の午後はあいていますか。
kyoo no gogo wa aiteimasuka
(Are you free this afternoon ?)

II: List of 25 Sample Ill-formed Sentences

Note: 、(in Japanese) and {..} (In English) represents a short term hesitation. {...} represents a long term hesitation. English translation is well-formed.

1. えっと来週の、えっと火曜日から、木曜日まで、名古屋に出張します。
   {etto} raishuu no {.. etto} kayoobi kara {..} mokuyoobi made {..} nagoya ni shuchoo shimasu
   (I will make a business trip to Nagoya from Tuesday to Thursday next week.)

2. えー、今週の金曜日に休暇をとりたいんですが。
   {ee..} koNshuu no kiNyoobi ni kyuuka o tori taiNdesuga
   (I would like to take a holiday this Friday.)

3. じゃあ、やめます。
   jaa {..} yame masu
   (Then, I'll cancel the request.)

4. 火曜日の、午後4時から... 講義を、入れたい。
   kayoobi no {..} gogo yoji kara {...} kougi o {..} ire tai
   (I have a lecture at 4 P.M. on Tuesday.)

5. えーと、来週の月曜日、えー 午前8時から... えー セミナーを、登録して下さい。

   {eeto ..} raishuu no getsuyoobi {.. ee} gozeN hachiji kara {... ee} seminaa o {..} tooroku shitekudasai
   (Please input the seminar at 8 A.M. next Monday.)

6. 金曜日、の午後4時から、テニスをします。
   kiNyoobi {..} no gogo yoji kara {..} tenisu o shimasu
   (I will play tennis from 4 P.M. next Friday.)

7. 近衛グラウンドで... お願いします。
   konoegurauNdo de {...} onegaishimasu
   (At Konoe ground, please.)

8. え 月曜日の勉強会を、午後2時からに、変更して下さい。
   {e} getsuyoobi no beNkyookai o {..} gogo niji kara ni {..} heNkoo

shitekudasai
(Please change the time of the next Monday's seminar to 2 P.M.)

9. 来週の一出張を、取り消します。
raishuu no {...} shuchoo o {..} torikeshimasu
(Cancel the business trip next week.)

10. 来週の火曜日の、テニス、の時間を変更します。
raishuu no kayoobi no {..} tenisu {..} no jikaN o heNkoo shimasu
(Change the time for playing tennis next Tuesday.)

11. え、1時からに変更して下さい。
{e ..} ichiji kara ni heNkoo shitekudasai
(Please change it to 1 o'clock.)

12. あのー、明日の会議の場所を変更します。
{anoo ..} asu no kaigi no basho o heNkoo shimasu
(Change the place of tomorrow's meeting.)

13. えー 大会議室で行います。
{ee} daikaigishitsu de okonai masu
(At the large meeting room.)

14. あのー、30日の会議はどこで。
{anoo ..} saNjuunichi no kaigi wa dokode
(Where will the meeting on the 30th be held ?)

15. え、テニスの予定はいつでしたか。
{e ..} tenisu no yotee wa itsu deshitaka
(On which day is tennis scheduled ?)

16. と、そのテニス、の予定を中止して下さい。
{to ..} sono tenisu {..} no yotee o chuushi shitekudasai
(Please cancel the tennis.)

17. えー 月曜日の、講義を、休講にします。
{ee} getsuyoobi no {..} kougi o {..} kyuukoo ni shimasu
(Cancel the next Monday's lecture.)

18. 金曜日には 何が 予定が入っていましたか。
    kiNyoobi niwa naniga {..} yotee ga haite imashitaka
    (What are the plans for next Friday ?)

19. あさってのセミナー、5時からでしたか。
    asate no seminaa {..} goji kara deshitaka
    (Will the seminar on the day after tomorrow start at 5 o'clock ?)

20. えー 何時からならあいていますか。
    {ee} naNji kara nara aiteimasuka
    (What time are you free ?)

21. 10時、からです。
    juuji {..} kara desu
    (After 10 o'clock.)

22. えー、何時までですか。
    {ee ...} naNji made desuka
    (Until what time ?)

23. 4日の講義を えー 中止します。
    yoka no kougi o {ee} chuushi shimasu
    (Cancel the lecture on 4th.)

24. え 来週の火曜日のテニスを、えー会議に変更します。
    {e} raishuu no kayoobi no tenisu o {.. ee} kaigi ni heNkoo shimasu
    (Cancel next Tuesday's tennis, and have a meeting instead.)

25. 午前9時から、12時まで。
    gozeN kuji kara {..} juuniji made
    (From 9 to 12 in the morning.)